



Infor Factory Track Extensibility Guide for CSI

Copyright © 2017 Infor

Important Notices

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

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

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

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

Trademark Acknowledgements

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

Publication Information

Release: Infor Factory Track Release level 6.01

Publication date: August 18, 2017

Table of Contents

About this guide	5
Organization.....	5
Contacting Infor.....	5
Extending Factory Track to utilize SyteLine UETs	7
Requirement	7
SyteLine Item Table UET	7
Factory Track changes	7
FT Database level.....	7
FT IDO level	7
FT forms level.....	9
Extending Factory Track to update extra information in SyteLine	10
Requirement	10
SyteLine Table and IDOs	10
SyteLine Table.....	10
SyteLine Stored Procedure.....	11
SyteLine IDO	12
SyteLine IDO Method	13
Factory Track changes	14
IDO Extension Class.....	14
Factory Track Pick N Ship Form Changes	17
Extending Factory Track to print additional information on the Barcode label.....	19
Requirement	19
Label printing process in Factory Track	19
Create records in Label printing tables	20
Label printing additional information	20
Using AES System to build customization	20
Creating Event Actions:	22
Example stored procedure creation:	25
Testing the customization	27

About this guide

This document is intended to provide information on extending the functionality of Infor Factory Track to meet some of the customization requirements. The document also addresses some common customization scenarios and provides explanation using examples.

Organization

This table shows the chapters of the guide:

Section	Description
Extending Factory Track to utilize SyteLine UET's	Ability to add UET's (User Extended Table Fields) to existing tables
Extending Factory Track to update extra information in SyteLine	Extending Factory Track to update extra information in SyteLine
Extending Factory Track to print additional Information on the Barcode Label	Explains the case where additional information must be printed

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal.

If we update this document after the product release, we will post the new version on this website. We recommend that you check this website periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Extending Factory Track to utilize SyteLine UETs

The Mongoose platform provides an option to add UETs (User Extended Table Fields) to existing tables. Some of the UET must be added to SyteLine tables. The example explains the changes that can be made to Factory Track utilizing the UET created in SyteLine.

Requirement

For example, a new field is added to the Item table to capture the default number of items in a container and this information must be displayed in the Factory Track transactions.

SyteLine Item Table UET

For example, a new UET uf_item_qty is added to the Item master table. This UET is defined to store the specified default number of items that can fit inside a container.

In SyteLine, this UET is automatically available in SLItems IDO and can be added to SyteLine forms.

Factory Track changes

The UET created in SyteLine must be available at the Factory Track database level but not at the IDO level.

FT Database level

If a query is made for 'slitem' view at the Factory Track database, the field uf_item_qty must be available. Stored procedures or functions developed can utilize this new field.

FT IDO level

As the UET is created on the SyteLine side, this UET is not available on the Factory Track side. To make this available in FT:

- 1 Create New IDO ICSSLItems_Ext as Extension of ICSSLItems.
- 2 Select Extend and Replace and check the other option to replace ICSSLItems IDO.

IDOs

IDO Name	New IDO	New Table	New Property	New Method
ICSLItems_Ext	IDO Name: ICSLItems_Ext Project Name: SLMaterial Description: Label String ID: oslItems		Extends: ICSLItems <input checked="" type="checkbox"/> Replace IDO Assembly Name: Ext Class Name:	Revision Num: 1 Revision Date: 6/5/2017 Locked By: sa Access As: MES

- 3 Add a new property to ICSLItems_Ext IDO for the UET Property created in SyteLine. To do so:
 - a Specify Uf_Item_Qty as the name of the property.
 - b Select Bound to Column property type as the field which is available in SLItem view at the Factory Track database level.
 - c Clear IDO cache using Configuration Manager or by restarting the IDO runtime.

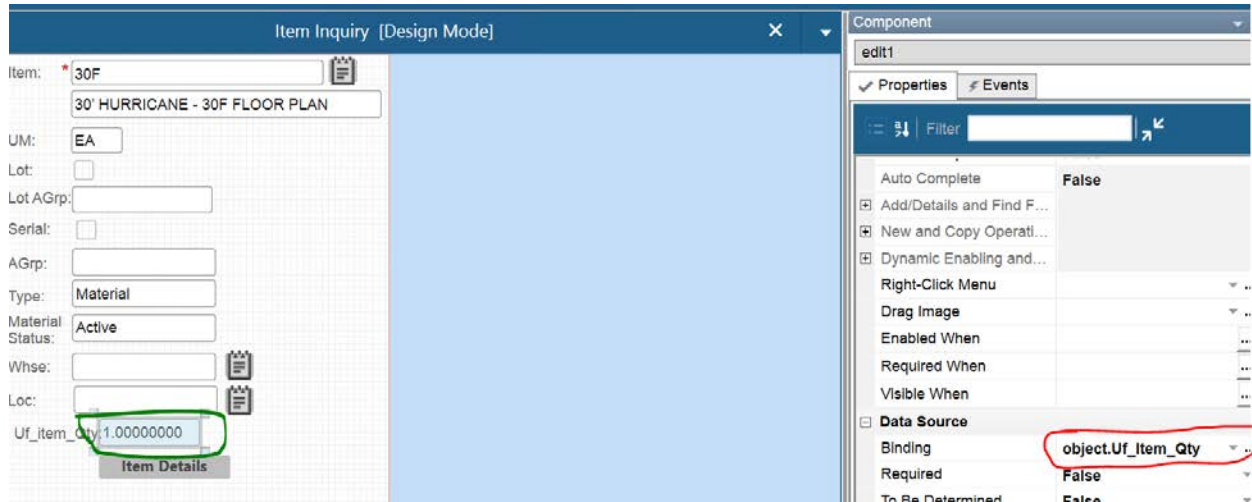
New Property...

IDO Name:	ICSLItems_Ext	Column Table Alias:	itm
Property Name:	Uf_Item_Qty	Column Name:	Uf_Item_Qty
Description:		Sequence:	235 <input type="checkbox"/> Pseudo Key
Property Type:	Bound to Column		
Property Class:			

[Property Overrides](#)
[Implementation](#)
[Subcollection](#)

FT forms level

The new field that is added to the Factory Track IDO can now be added to the Factory Track forms. For example, add this field to the Item Inquiry Transaction form:



- 1 Add an Edit Field to Item Inquiry form.
- 2 Set the Binding field to Uf_Item_Qty Property.

Extending Factory Track to update extra information in SyteLine

Most of the Factory Track transactions post transactional update to SyteLine. This section explains extending Factory Track to update the additional information in SyteLine.

Requirement

For example; ShipmentTracking table (customization table) must be updated in SyteLine as part of Pick N Ship transaction form in Factory Track.

SyteLine Table and IDOs

The table definition for Shipment Tracking with IDO and an IDO method (type=stored procedure), to capture the information.

SyteLine Table

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[ShipmentTracking_Test](
    [co_num] [dbo].[CoNumType] NULL,
    [co_line] [dbo].[CoLineType] NULL,
    [Datetime] [dbo].[DateType] NULL,
    [Sequence] [int] IDENTITY(1,1) NOT NULL,
    [track_num] [nvarchar](200) NULL)
```


SyteLine Stored Procedure

```
CREATE PROCEDURE [dbo].[AddTrackNumberSp](
    @CoNum          CoNumType
    , @CoLine       CoLineType
    , @TrackNum      nvarchar(200)
)
AS
Begin
    INSERT INTO ShipmentTracking_Test(
        co_num
        , co_line
        , Datetime
        , track_num
    )
    VALUES(
        @CoNum
        , @CoLine
        , GETDATE()
        , @TrackNum)
END
```

SyteLine IDO

New IDO **New Table** **New Property** **New Method**

IDO Name: Extends: Revision Num:

Project Name: Replace Revision Date:

Description: IDO Assembly Name: Locked By:

Label String ID: Ext Class Name: Access As:

Caption: Ext Class Namespace:

Outtrigger Profile:

- Tables** **Properties** **Methods** **Filters** **Referenced Property Classes** **Advanced Attributes**

Tables Properties Methods Filters Rules

	Table Name	Table Alias	Table Type	Join Type	Join Text
1 ▶	ShipmentTracking_Test	trk	Primary Base	Inner	

SyteLine IDO Method

Method Attributes

IDO Name: SLShipmentTracking

Method Name: AddTrackNumberSp

Description:

Method Type: Stored Procedure - Standard Method

Stored Procedure: AddTrackNumberSp

New Method...

Resync Parameters

Parameters

	Parameter Name	Sequence	User Data Type	Data
1 ▶	CoNum	1	VARIANT	VA
2	CoLine	2	VARIANT	SH
3	TrackNum	3	VARIANT	VA
*				

Factory Track changes

Factory Track utilizes the Mongoose IDO communication technology to invoke SyteLine IDOs to process transactions in SyteLine. To perform this activity:

- 1 Create a new IDO Extension class and include the logic to invoke SyteLine IDO.
- 2 Create a new Factory Track IDO/IDO Method that references this IDO Extension class.
- 3 Factory Track forms can now invoke this new IDO Method to perform this additional update.

IDO Extension Class

ICERPLayer Extension class is a core Factory Track assembly that communicates with SyteLine. This assembly has many reusable methods. The process:

- 1 Create a new Extension class project using Visual Studio (class library project).
- 2 Add reference dlls. Ensure that the ICERPLayer dll reference is added to the project.
- 3 Create a new class (either vb.net or c#) which class must extend ICERPLayer.ErpInteractionBase class. Part of this class create a new method.

Note: ExecuteERPUpdateRequest is the method responsible for invoking the SyteLine IDO methods.



- 4 Import this Extension class library into Factory Track.
- 5 Create a new IDO which has reference to this Extension class library.

[New IDO](#) [New Table](#) [New Property](#) [New Method](#)

IDO Name: Extends: Revision Num:

Object Name: Replace Revision Date:

Description: IDO Assembly Name: Locked By:

String ID: Ext Class Name: Access As:

Caption: Ext Class Namespace:

User Profile:

- [Tables](#) [Properties](#) [Methods](#) [Filters](#) [Referenced Property Classes](#) [Advanced Attributes](#)

[Tables](#) [Properties](#) [Methods](#) [Filters](#) [Rules](#)

Table Name	Table Alias	Table Type	Join Type	Join Text
AccessAs	acs	Primary Base	Inner	

6 Create a new IDO Method which can access the Extension class method.

Method Attributes

IDO Name:

Method Name:

Description:

Method Type: *

Stored Procedure:

Parameters

	Parameter Name	* Sequence	* User Data Type	Da
1 ▶	CoNum	1	VARIANT	VA
2	CoLine	2	VARIANT	VA
3	TransNum	3	VARIANT	VA
*				

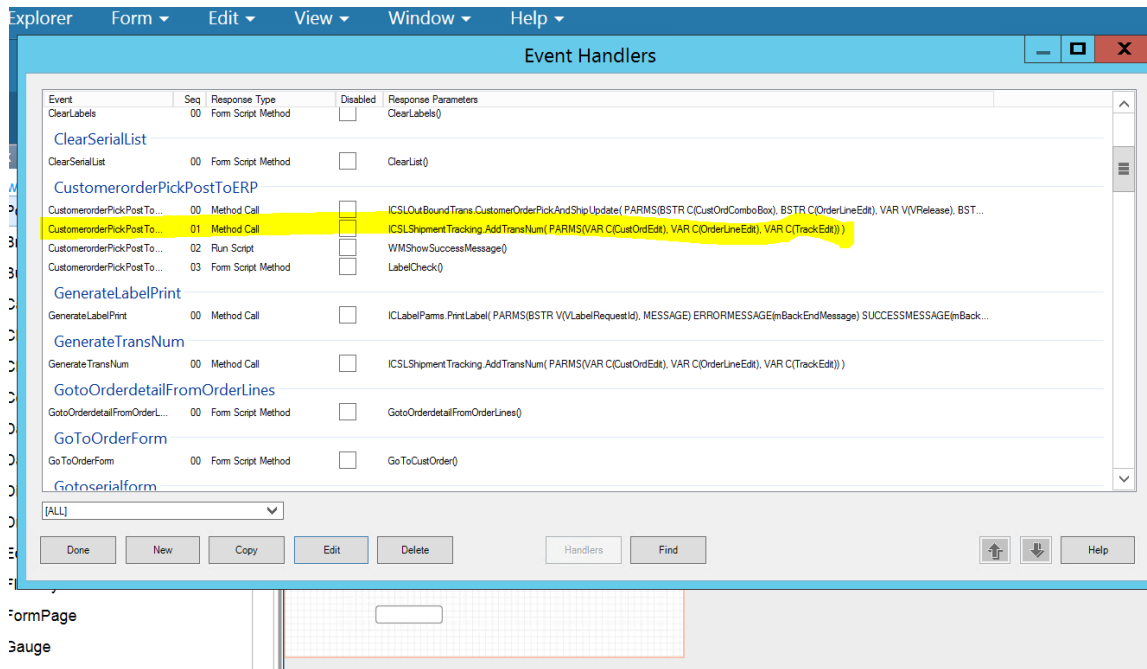
Factory Track Pick N Ship Form Changes

The Pick N Ship transaction form can be modified to invoke this new IDO method by specifying the required inputs. The process:

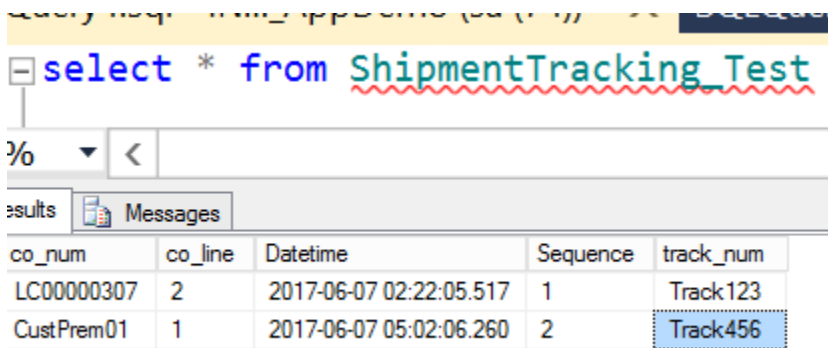
- 1 Add a new field Tracking Number to the Pick N Ship Transaction form.

The image shows a screenshot of a web form for a Pick N Ship transaction. The form is set against a light gray grid background. It contains several input fields and buttons. The fields are: Order (text), Line (text), Item (text), Required (text and checkbox), Loc (text with a red asterisk and a calendar icon), Lot (text with a calendar icon), Qty (text, checkbox, and calendar icon), and a newly added Tracking Number field (text) which is highlighted with a yellow background. At the bottom of the form are two buttons: 'Back' and 'Process'.

- 2 Add a Method Call, after the Pick N Ship update, to invoke the IDO method created.



Note: When the Pick N Ship transaction is performed, the tracking number on the form must be updated in the Shipment Tracking table in SyteLine.



Extending Factory Track to print additional information on the Barcode label

Factory Track provides varied information that can be printed on a Barcode label. This information is captured in the label_print_dtl table. This section explains the scenarios wherein additional information must be printed.

Requirement

For example; Print Item Unit Weight and Item Lead Time on the Barcode label.

Label printing process in Factory Track

Some of the transactions in Factory Track can be configured to print labels. The process to print the labels :

- 1 Create records in the Label printing tables in the Factory Track database.
- 2 Create a file with instructions and data from the Label printing tables, so that the Bartender application can print the label to the label printer.

Note: Based on the type of installation of Factory Track, (On Premises/Cloud), the above process can be configured in Factory Track using the Label Interface form.

Interface Nu...	
1	BarTender

Interface Number: BarTender

Interface Type: File

File Path: C:\labels\output

Field Separator: |

Label Template Path: C:\Program Files (x86)\Infor\FactoryTrack

- 3 Specify the information for the two Interface Type scenarios:
 - a Set Interface Type to File for On Premises installation.
 - b Set Interface Type to Database Only for Cloud installation.

Create records in Label printing tables

As part of the Label printing process, Factory Track creates data in these tables in the Factory Track application database:

- `ic_label_print_history` - Contains data to provide Bartender specific instructions (example number of copies, format name, printer name etc.)
- `ic_label_print_dtl` - Contains the actual data to be printed (example item, warehouse, order number etc.)

The data in these tables is used to print the file with data and instructions.

Currently, the `ic_label_print_dtl` table contains about close to 90+ different fields of information that can be printed on the label. This meets most of the customer label printing data requirements.



```
File Edit Format View Help
11Miscreceipt.dd - Notepad
File Edit Format View Help
%BTW% /AF="C:\Program Files (x86)\Infor\FactoryTrack\item.btw" /PRN="\\Bullzip PDF Printer" /P /D="%trigger file Name%" /C-1 /R-3
%END%
Bartender Instruction Data
LabelRequestIDString|LabelRequestID|LabelRequestSeqString|LabelRequestSeq|SiteString|Site|WhseString|Whse|LocString|Loc|ItemString|Item|DescriptionString|Description|
QtyString|Qty|UMString|UM|RevisionString|Revision|LotString|Lot|SerNumString|SerNum|ProjectString|Project|OrderTypeString|OrderType|OrderNumberString|OrderNumber|
OrderLineString|OrderLine|OrderReleaseString|OrderRelease|CustVendorString|CustVendor|CustVendorItemString|CustVendorItem|OrderUMString|OrderUM|Gs1CompanyCodeString|
Gs1CompanyCode|SerialShippingContainerCodeString|SerialShippingContainerCode|GlobalTradeItemNumberString|GlobalTradeItemNumber|ReceiptNumString|ReceiptNum|ShipmentString|
Shipment|ShipmentPkgString|ShipmentPkg|CarrierNameString|CarrierName|CarrierPkgString|CarrierPkg|ShipViaDescriptionString|ShipViaDescription|WeightString|Weight|
ProductionLineString|ProductionLine|CrossdockLocationString|CrossdockLocation|CrossdockOrderString|CrossdockOrder|RefOrderTypeString|RefOrderType|RefOrderNumberString|
RefOrderNum|RefOrderLineString|RefOrderLine|RefOrderReleaseString|RefOrderRelease|MfgDateString|MfgDate|ExpiryDateString|ExpiryDate|ContainerNumString|ContainerNum|
NumberOfBoxesString|NumberOfBoxes|AltItemCodeString|AltItemCode|ItemTypeString|ItemType|DescriptionBString|DescriptionB|DescriptionCString|DescriptionC|DescriptionDString|
DescriptionD|LocFromString|LocFrom|OperationString|Operation|QtyAdvisedString|QtyAdvised|RunNumberString|RunNumber|SearchKeyString|SearchKey|SequenceString|Sequence|
SerNumString|SerNum|DateString|Date|HULevelString|HULevel|MultiItemHUString|MultiItemHU|ParentHUString|ParentHU|QtyStorageUnitString|QtyStorageUnit|ReceiptLineString|ReceiptLine|
SerialPresentString|SerialPresent|ShipmentLineString|ShipmentLine|StatusString|Status|FixedLocString|FixedLoc|HeightString|Height|InvDateString|InvDate|InspectionFlagString|
InspectionFlag|LengthUnitString|LengthUnit|LotCertString|LotCert|RecommendedLocString|RecommendedLoc|SupplierLotString|SupplierLot|WidthString|Width|DepthString|Depth|
WeightUnitString|WeightUnit|WorkCenterString|WorkCenter|ActivityDescString|ActivityDesc|CostCompDescString|CostCompDesc|ProjectDescString|ProjectDesc|ElementDescString|ElementDesc|
ExtensionDescString|ExtensionDesc|FromActivityString|FromActivity|CostComponentString|CostComponent|ElementString|Element|ExtensionString|Extension|ReceiptDateString|
ReceiptDate|KanbanIDString|KanbanID|KanbanLocString|KanbanLoc|ReplenishWhseString|ReplenishWhse|QtyOpenString|QtyOpen|DayString|Day|MonthString|Month|YearString|Year|
Add1String|Add1|Add2String|Add2|CustVendNameString|CustVendName|UserNameString|UserName|LabelName|NumCopies|PrinterName|RefRequestID|UserID|LabelPrintSoftware|FileName|
QuietPrint|TaskTypeString|TaskType|TaskCodeString|TaskCode|TaskDescriptionString|TaskDescription|EmployeeNumberString|EmployeeNumber|FirstNameString|FirstName|LastNameString|
LastName|BadgeNumberString|BadgeNumber|ReasonCodeString|ReasonCode|ReasonCodeDescriptionString|ReasonCodeDescription|
Header Row of Fields
Label Request ID||Label Request Seq||Site|Whse|Main|Location|STOCK|Item|FA-10000|Description|B|C|E, Model-30, 26 |Qty|10.00000000|UM|EA|Revision||Lot|4|Serial
Number||Project||Order Type||Order Number||Order Line||Order Release||Cust Vendor Item||Cust Vendor Item||Order UM||Gs1 Company Code||Serial Shipping Container Code||Global Trade
Item Number||Receipt Num||Shipment||Shipment Pkg||Carrier Name||Carrier Pkg||Ship Via Description||Weight||Production Line||Cross dock Location||Cross dock Order||Ref Order
Type||Ref Order Num||Ref Order Line||Ref Order Release||Mfg Date||Expiry Date||Container Num||Number of Boxes|1.00000000|Alternate Item|Item Type||Description B||Description
C||Description D||Location From||Operation||Qty Advised||Run Number||Search Key||Sequence|Serial Number||Date||HU Level||Multi-Item HU||Parent HU||Qty Storage U/M||Receipt
Line||Serial Present||Shipment Line||Status||Fixed Location||Height||Inv Date||Inspection Flag||Length U/M||Lot Certification||Recommended Loc||Supplier Lot||Width||Depth||
Height U/M||Work Center||Activity Description||Cost Component Description||Project Description||Element Description||Extension Description||From Activity||Cost Component||
Element||Extension||Receipt Date||Kanban ID||Kanban Location||Replenish Warehouse||Qty Open||Day||Month||Year||Address (1)||Address (2)||Customer/Vendor Name||User Name||
Item.btw|1|\\Bullzip PDF Printer||171|1|Miscreceipt.dd|1|Task Type||Task Code||Task Description||Employee Number||First Name||Last Name||Badge Number||Reason Code|Reason Code
Description||
Actual Data for Label
```

Label printing additional information

To support the option where customers require additional information to be printed in the label more than what is currently available, Factory Track provides an extra table called 'ic_label_print_extension'.

Customers can create a small customization that can insert data in this 'ic_label_print_extension' table. Factory Track must add this additional data to the file created for BarTender, as part of creating the file process.

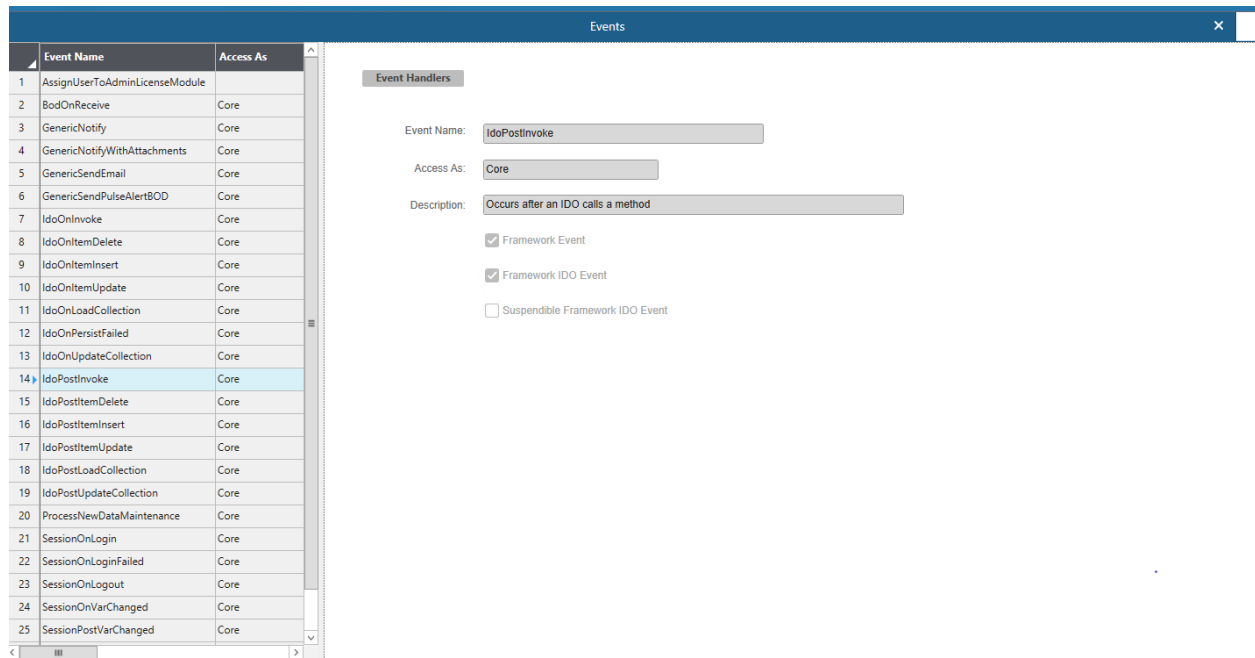
Using AES System to build customization

AES (Application Event System) provides an easy option enabling quicker customization by inserting data in 'ic_label_print_extension' table.

Factory Track invokes the 'ICInsertLabelPrintDetailsSp' method to insert data in the Label Print tables. AES invokes 'IdoPostEvent' after this method is executed. As part of the customization a new event handler can be added to 'IdoPostEvent' that can insert additional data in the 'ic_label_print_extension' table.

To create the event handler and associated code:

- 1 Access the Events form
- 2 Specify 'IdoPostInvoke' in the Event Name field.



	Event Name	Access As
1	AssignUserToAdminLicenseModule	
2	BodOnReceive	Core
3	GenericNotify	Core
4	GenericNotifyWithAttachments	Core
5	GenericSendEmail	Core
6	GenericSendPulseAlertBOD	Core
7	IdoOnInvoke	Core
8	IdoOnItemDelete	Core
9	IdoOnItemInsert	Core
10	IdoOnItemUpdate	Core
11	IdoOnLoadCollection	Core
12	IdoOnPersistFailed	Core
13	IdoOnUpdateCollection	Core
14	IdoPostInvoke	Core
15	IdoPostItemDelete	Core
16	IdoPostItemInsert	Core
17	IdoPostItemUpdate	Core
18	IdoPostLoadCollection	Core
19	IdoPostUpdateCollection	Core
20	ProcessNewDataMaintenance	Core
21	SessionOnLogin	Core
22	SessionOnLoginFailed	Core
23	SessionOnLogout	Core
24	SessionOnVarChanged	Core
25	SessionPostVarChanged	Core

Event Handlers

Event Name:

Access As:

Description:

Framework Event

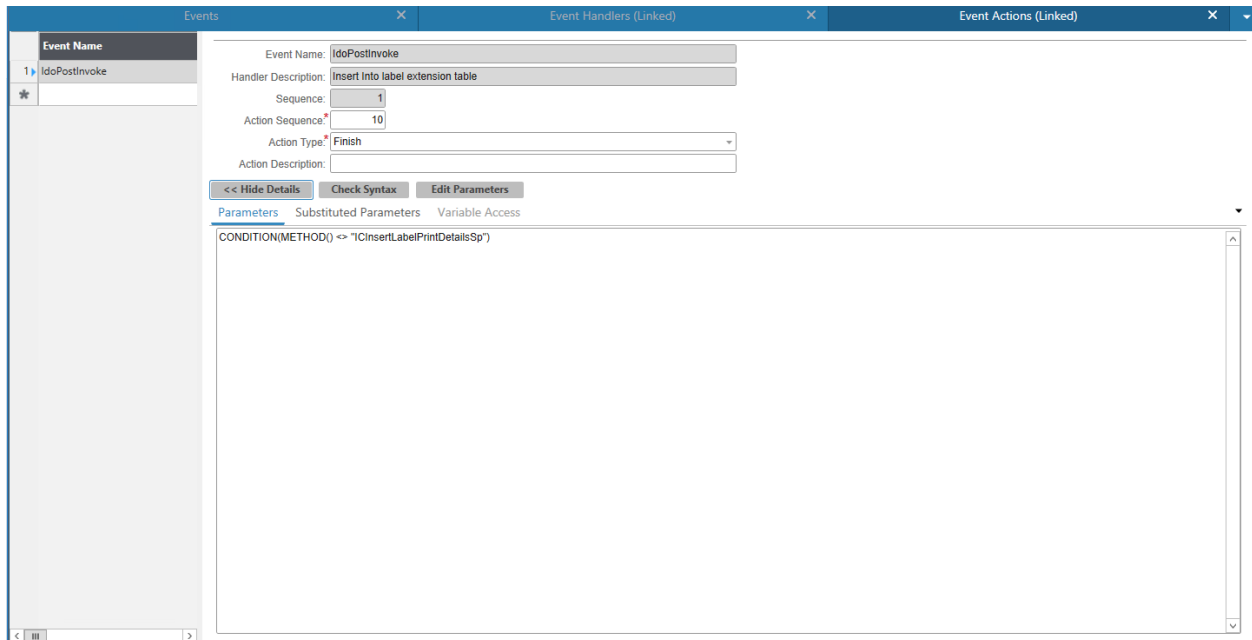
Framework IDO Event

Suspendible Framework IDO Event

- 3 Click Event Handler for the 'IdoPostInvoke' event. The Event Handler (linked) form for the 'IdoPostInvoke' event is displayed.
- 4 Specify this information to create a new Event Handler:
 - a Specify 'ICLabelPrintDtIs' in the **Applies to Objects** field.
 - b Specify description in the **Handler Description** field.
- 5 Click Save.

Creating Event Actions:

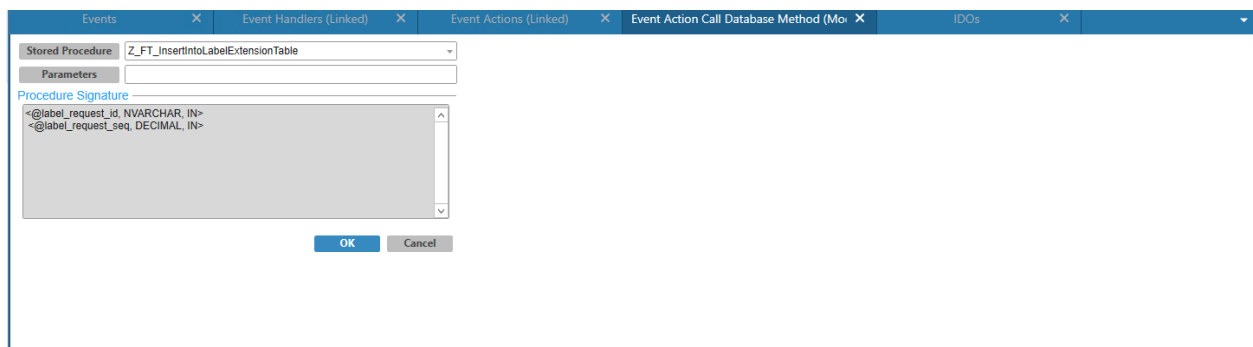
- 1 Click **Event Actions** on the Event Handlers form. The Event Actions (Linked) form is displayed.



The screenshot shows the 'Event Actions (Linked)' form. The 'Event Name' is 'IdoPostInvoke'. The 'Handler Description' is 'Insert into label extension table'. The 'Sequence' is '1' and the 'Action Sequence' is '10'. The 'Action Type' is 'Finish'. The 'Action Description' is empty. Below the form, there are buttons for '<< Hide Details', 'Check Syntax', and 'Edit Parameters'. The 'Parameters' tab is selected, showing a text area with the content 'CONDITION(METHOD) <=> 'ICInsertLabelPrintDetailsSp''.

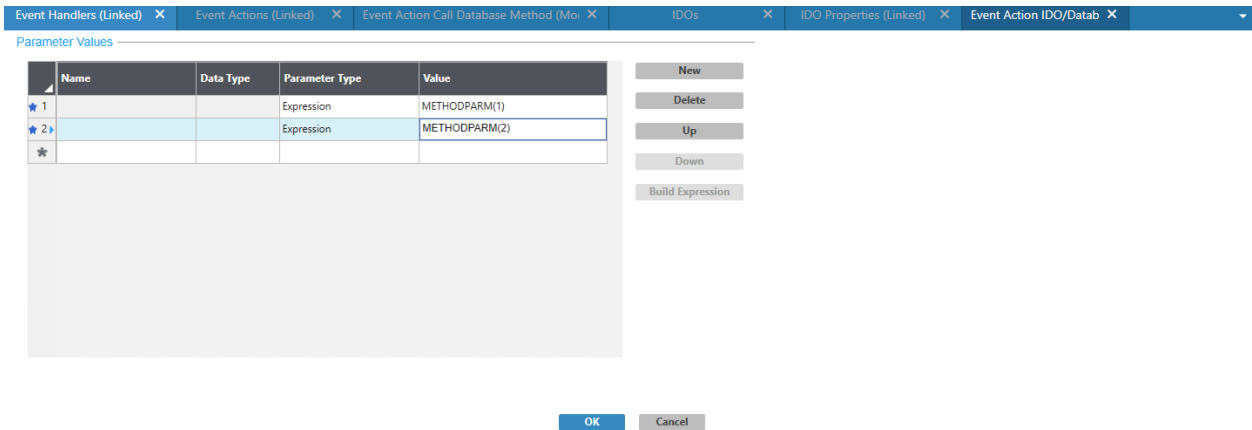
Note: This event creation process is completed only if the **Method () <=>** is set to 'ICInsertLabelPrintDetailsSp'. The Factory Track program must use 'ICInsertLabelPrintDetailsSp' method to insert data in the Label Print tables, to ensure that the logic is executed only for this method.

- 2 Create a new Event Action which must execute a stored procedure using which logic can be developed to retrieve additional set of data from ERP and populate the 'ic_label_print_extension' table. Specify this information:
 - a Set the Action Type field to **Call Database Method**, (for example)
 - b Click **Edit Parameters**.



The screenshot shows the 'Event Action Call Database Method (Mo)' dialog box. The 'Stored Procedure' field is set to 'Z_FT_InsertIntoLabelExtensionTable'. The 'Parameters' field is empty. The 'Procedure Signature' field shows '<@label_request_id, NVARCHAR, IN>' and '<@label_request_seq, DECIMAL, IN>'. There are 'OK' and 'Cancel' buttons at the bottom.

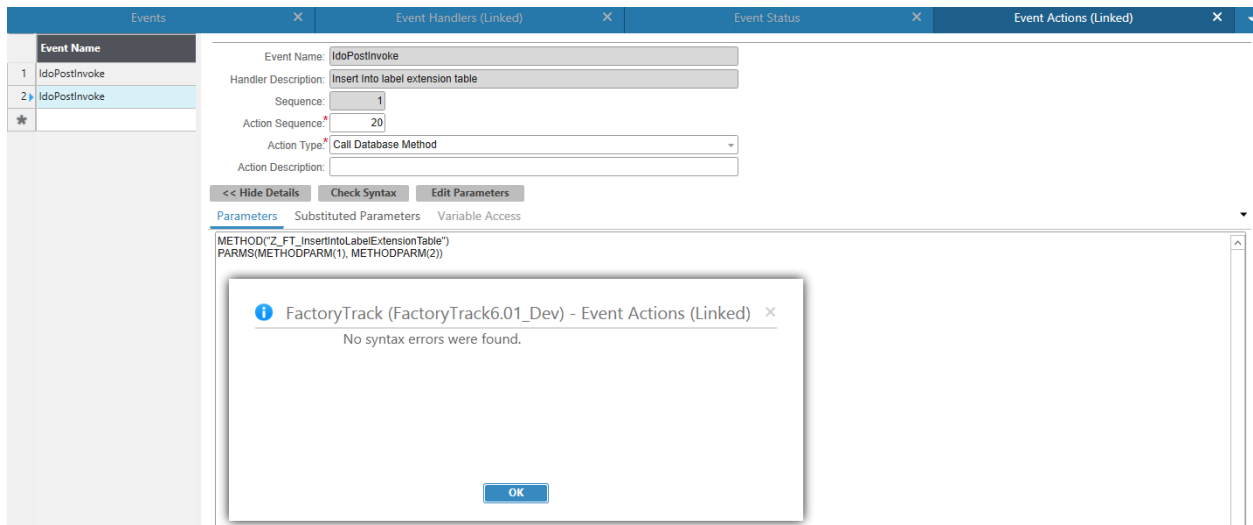
- c Specify the **Stored Procedure** name which contains the logic to insert additional label data. This stored procedure requires inputs, which can be used to query and populate the additional data. This input must be specified to the label data that is attached.
- d Click **Parameters** to provide this information:
 - MethodParam(1) is label request id
 - MethodParam(2) is label request sequence.



Note: In this case, MethodParam(1) is the first input provided to the 'ICInsertLabelPrintDetailsSp' method, by the Factory Track programs while inserting data in the Label Print tables.

See IDO methods for 'ICInsertLabelPrintDetailsSp' for the list of inputs that can be provided for this method.

- 3 The Event Action screen is displayed.



- 4 Click Check Syntax for each of the event action to ensure that there are no syntax errors.

Example stored procedure creation:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF OBJECT_ID('dbo.Z_FT_InsertIntoLabelExtensionTable') IS NOT NULL
    DROP PROCEDURE dbo.Z_FT_InsertIntoLabelExtensionTable
GO

CREATE PROCEDURE [dbo].[Z_FT_InsertIntoLabelExtensionTable] (
    @label_request_id    ILabelRequestIdType,
    @label_request_seq    SequenceType
) AS

DECLARE
    @Severity            INT,
    @Item                LblItemType,
    @ItemLeadTime        LeadTimeType,
    @UnitWeight          ItemWeightType

SET @Severity = 0

--Query ic_label_print_dtl table to retrieve item used during label printing
SELECT    @Item=item from ic_label_print_dtl
WHERE     label_request_id    = @label_request_id
AND       label_request_seq   = @label_request_seq

--Query slitem view to get the lead time for @Item
SELECT    @ItemLeadTime=lead_time,@UnitWeight=unit_weight from
slitem
WHERE     item                = @Item
```

```
--insert into ic_label_print_extension table
insert into ic_label_print_extension (label_request_id,
label_request_seq,label_request_sub_seq,field_name,field_value)
values(@label_request_id, @label_request_seq, '1','Item Lead
Time',@ItemLeadTime);

insert into ic_label_print_extension (label_request_id,
label_request_seq,label_request_sub_seq,field_name,field_value)
values(@label_request_id, @label_request_seq, '2','Item Unit
Weight',@UnitWeight);

RETURN @Severity

Go
```


Testing the customization

To test the customization:

The image displays two views of the 'Stock Move' application interface. The left view is the main data entry form, and the right view is a detailed printer configuration panel.

Main Data Entry Form (Left):

- Whse:** MAIN
- Container:** (empty)
- Item:** 908 (Gear, 8 Tooth)
- Location:** STOCK
- Document:** (empty)
- On Hand:** 2.00
- Available:** 2.00 EA
- To Move:** 1.00 EA
- To Location:** BIKE 1.0
- Process** button

Printer Configuration Panel (Right):

- Printer:** TranPrinter
- Label Per Box:** 1
- Qty Remaining:** 0.00
- No Of Boxes:** 1
- Qty Per Box:** 1.00
- Buttons:** Back, Clear Labels, Print

- 1 Execute a simple Stock Move transaction.
- 2 The report for the label printed, must include the additional data (Item Lead Time and Unit Weight) from Item 908 in the 'ic_label_print_extension' table and the data must be part of label print file for the BarTender print service.

```

select * from ic_label_print_extension
|
select * from ic_label_print_dtl order by createDate desc

```

label_request_id	label_request_seq	label_request_sub_seq	field_name	field_value	NoteExistsFlag	RecordDate	RowPointer	CreatedBy
13	1	1	Item Lead Time	0	0	2016-10-22 18:07:12.070	525D71FC-E352-403F-A94D-5FB65285783E	kmadduri
13	1	2	Item Unit Weight	0.00000	0	2016-10-22 18:07:12.070	255AC45B-6311-4DDA-838B-28F1BFA72367	kmadduri

label_request_id	label_request_seq	site	whse	loc	item	description	number_of_boxes	qty	u_m	revision	lot	ser_num	project	created_date
13	1	LA	MAIN	BIKE 10	908	Gear, 8 Tooth	1.00000000	1.00000000	EA	1	NULL	NULL	NULL	2016-10-22 18:07:12.070
12	1	LA	MAIN	BIKE 10	908	Gear, 8 Tooth	1.00000000	1.00000000	EA	1	NULL	NULL	NULL	2016-10-22 18:07:12.070
11	1	LA	MAIN	STOCK	FA-10000	Bicycle, Model-30, 26"	1.00000000	10.00000000	EA	1	4	NULL	NULL	2016-10-22 18:07:12.070
10	1	LA	MAIN	FIN-1	FA-10000	Bicycle, Model-30, 26"	1.00000000	10.00000000	EA	1	4	NULL	NULL	2016-10-22 18:07:12.070
9	2	LA	MAIN	STOCK	TC-10000	Tablet Computer with Smart Cover	1.00000000	1.00000000	EA	1	NULL	-2	NULL	2016-10-22 18:07:12.070
9	1	LA	MAIN	STOCK	TC-10000	Tablet Computer with Smart Cover	1.00000000	1.00000000	EA	1	NULL	-1	NULL	2016-10-22 18:07:12.070
8	2	LA	MAIN	STOCK	TC-10000	Tablet Computer with Smart Cover	1.00000000	1.00000000	EA	1	NULL	0	NULL	2016-10-22 18:07:12.070
8	1	LA	MAIN	STOCK	TC-10000	Tablet Computer with Smart Cover	1.00000000	1.00000000	EA	1	NULL	-1	NULL	2016-10-22 18:07:12.070
7	2	LA	MAIN	STOCK	TC-10000	Tablet Computer with Smart Cover	1.00000000	1.00000000	EA	1	NULL	0	NULL	2016-10-22 18:07:12.070
7	1	LA	MAIN	STOCK	TC-10000	Tablet Computer with Smart Cover	1.00000000	1.00000000	EA	1	NULL	-1	NULL	2016-10-22 18:07:12.070

Query executed successfully. usalvwfctrk61db (11.0 SP1) sa (61) FactoryTrack_App 00:00:00 21 rows

```

%&TH% /AF="C:\Program Files (x86)\Infor\FactoryTrack\item.btw" /PRN=""\Bullzip PDF Printer" /P /D="Trigger File Name%" /C=1 /R=3
%END%
LabelRequestIdString|LabelRequestSeqString|LabelRequestSeqString|LabelRequestSeqString|SiteString|SiteString|WhseString|WhseString|LocString|LocString|ItemString|ItemString|DescriptionString|DescriptionString|QtyString|QtyString|UMString|UMString|RevisionString|RevisionString|LotString|LotString|SerNumString|SerNumString|ProjectString|ProjectString|OrderTypeString|OrderTypeString|OrderNumberString|OrderNumberString|OrderLineString|OrderLineString|OrderReleaseString|OrderReleaseString|CustVendorString|CustVendorString|CustVendorItemString|CustVendorItemString|OrderUMString|OrderUMString|GslCompanyCodeString|GslCompanyCodeString|SerialShippingContainerCodeString|SerialShippingContainerCodeString|GlobalTradeItemNumberString|GlobalTradeItemNumberString|ReceiptNumString|ReceiptNumString|ShipmentString|ShipmentPkgString|ShipmentPkgString|CarrierNameString|CarrierNameString|CarrierPkgString|CarrierPkgString|ShipViaDescriptionString|ShipViaDescriptionString|WeightString|WeightString|ProductionLineString|ProductionLineString|CrossdockLocationString|CrossdockLocationString|CrossdockOrderString|CrossdockOrderString|RefOrderTypeString|RefOrderTypeString|RefOrderNumString|RefOrderNumString|RefOrderLineString|RefOrderLineString|RefOrderReleaseString|RefOrderReleaseString|MfgDateString|MfgDateString|ExpiryDateString|ExpiryDateString|ContainerNumString|ContainerNumString|NumberOfBoxesString|NumberOfBoxesString|AltItemCodeString|AltItemCodeString|ItemCodeString|ItemCodeString|ItemDescriptionString|ItemDescriptionString|DescriptionString|DescriptionString|DescriptionString|DescriptionString|DescriptionString|DescriptionString|LocFromString|LocFromString|LocToString|LocToString|OperationString|OperationString|QtyAdvisedString|QtyAdvisedString|RunNumberString|RunNumberString|SearchKeyString|SearchKeyString|SequenceString|SequenceString|SernString|SernString|DateString|DateString|HULevelString|HULevelString|MultiItemHUString|MultiItemHUString|ParentHUString|ParentHUString|QtyStorageUnitString|QtyStorageUnitString|ReceiptLineString|ReceiptLineString|SerialPresentString|SerialPresentString|ShipmentLineString|ShipmentLineString|StatusString|StatusString|FixedLocString|FixedLocString|HeightString|HeightString|InvDateString|InvDateString|InspectionFlagString|InspectionFlagString|LengthUnitString|LengthUnitString|LotCertString|LotCertString|RecommendedLocString|RecommendedLocString|SupplierLotString|SupplierLotString|WidthString|WidthString|DepthString|DepthString|WeightUnitString|WeightUnitString|WorkCenterString|WorkCenterString|ActivityDescString|ActivityDescString|CostCompDescString|CostCompDescString|ProjectDescString|ProjectDescString|ElementDescString|ElementDescString|ExtensionDescString|ExtensionDescString|FromActivityString|FromActivityString|CostComponentString|CostComponentString|ElementString|ElementString|ExtensionString|ExtensionString|ReceiptDateString|ReceiptDateString|KanbanIDString|KanbanIDString|KanbanLocString|KanbanLocString|ReplenishWhseString|ReplenishWhseString|QtyOpenString|QtyOpenString|DayString|DayString|MonthString|MonthString|YearString|YearString|Add1String|Add1String|Add2String|Add2String|CustVendNameString|CustVendNameString|UserNameString|UserNameString|LabelNameString|LabelNameString|NumCopiesString|NumCopiesString|PrinterNameString|PrinterNameString|RefRequestIDString|RefRequestIDString|UserIDString|UserIDString|LabelPrintSoftwareString|LabelPrintSoftwareString|QuietPrintString|QuietPrintString|TaskTypeString|TaskTypeString|TaskCodeString|TaskCodeString|TaskDescriptionString|TaskDescriptionString|EmployeeNumberString|EmployeeNumberString|FirstNameString|FirstNameString|LastNameString|LastNameString|BadgeNumberString|BadgeNumberString|ReasonCodeString|ReasonCodeString|ReasonCodeDescriptionString|ReasonCodeDescriptionString|ItemLeadTimeString|ItemLeadTimeString|ItemUnitWeightString|ItemUnitWeightString
Unit Weight Data
Label Request ID|13|Label Request Seq|1|Site|LA|Whse|MAIN|Location|BIKE 10|Item|908|Description|Gear, 8 Tooth|Qty|1.00000000|UM|EA|Revision|1|Lot|Serial Number|Project|Order Type|Order Number|Order Line|Order Release|Cust Vendor Item|Order UM|Gsl Company Code|Serial Shipping Container Code|Global Trade Item Number|Receipt Num|Receipt Num|Shipment|Shipment Pkg|Carrier Name|Carrier Pkg|Ship Via Description|Weight|Production Line|Cross dock Location|Cross dock Order|Ref Order Type|Ref Order Num|Ref Order Line|Ref Order Release|Mfg Date|Expiry Date|Container Num|Number of Boxes|Alt Alternate Item|Item Type|Description B|Description C|Description D|Location From|Operation|Qty Advised|Run Number|Search Key|Sequence|Serial Number|Date|HU Level|Multi-Item HU|Parent HU|Qty Storage U/M|Receipt Line|Serial Present|Shipment Line|Status|Fixed Location|Height|Inv Date|Inspection Flag|Length U/M|Lot Certification|Recommended Loc|Supplier Lot|Width U/M|Work Center|Activity Description|Cost Component Description|Project Description|Element Description|Extension String|From Activity|Cost Component|Element|Extension|Receipt Date|Kanban ID|Kanban Location|Replenish Warehouse|Qty Open|Day|Month|Year|Address (1)|Address (2)|Customer/Vendor Name|User Name|Item.btw|1|\Bullzip PDF Printer|131|1|StockMove.dd|0|Task Type|Task Code|Task Description|Employee Number|First Name|Last Name|Badge Number|Reason Code|Reason Code Description|Item Lead Time|0|Item Unit Weight|0.00000

```