

Produmex Manufacturing Customization Guide

Customization Process

Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

The customization process for the Produmex mobile applications consists of the following steps:

- Enable the Customization Assist mode and run the client application. Check the customization information of the screen.
- Create the custom field and/or the SAP user query.
- Restart the client application and check the results of the customization.

Many screens of Produmex mobile applications are customizable with SAP Business One's queries. The used query have to be named specifically based on the screen it is used on. The query names, and parameters can be found with the help of the 'Customization Assist'. The query result column names indicate which fields to set.

Supported customization types:

- Preset field data
- Validation: The system checks if the entered values are correct during a 'validate' or a 'button' event.
- Add new fields/Hide existing fields

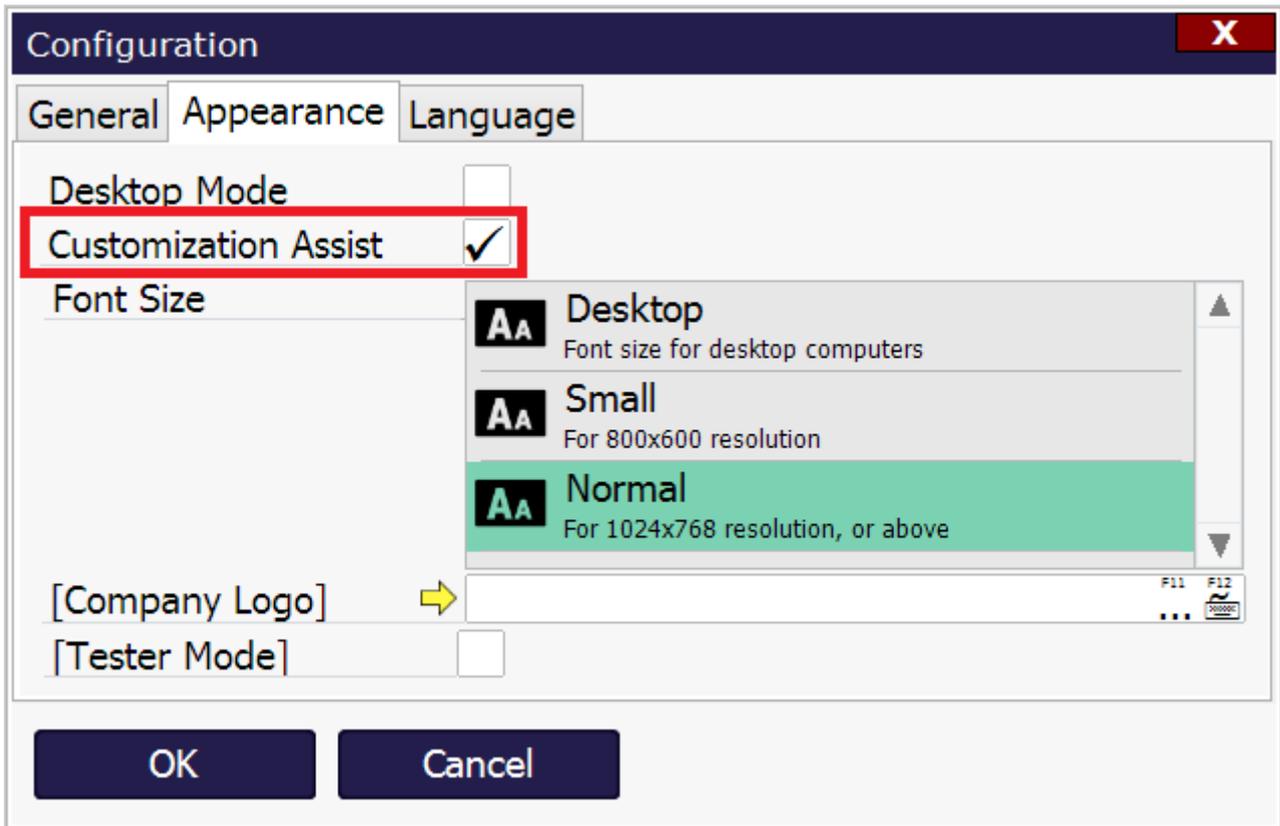
1. Configure and run client application

1.1. Enable the Customization Assist

To see information regarding the customization possibilities enable the Customization Assist mode.

The Customization Assist mode is only needed to see customization possibilities. It is not required to be enabled during normal operations.

Start the Produmex PDC Configuration application. On the 'Appearance' tab tick the Customization Assist checkbox to enable the Customization Assist mode.

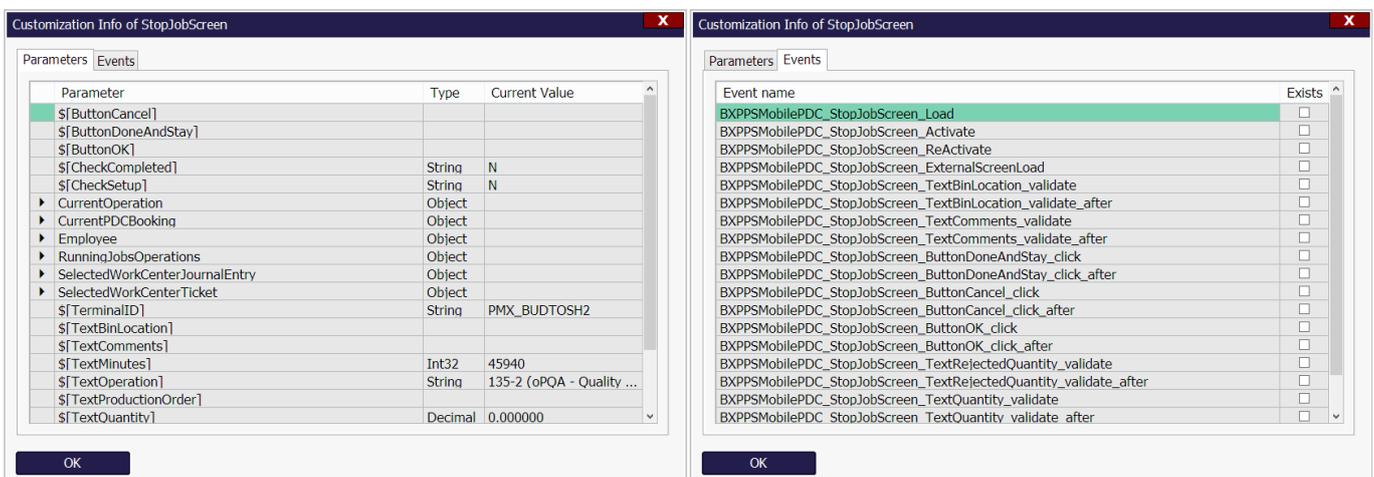


1.2. Run client application

After the Customization Assist Mode has been activated, run the Produmex client application. Go to the screen to be customized.

Press the  button on the top of the screen. The 'Customization Info of ScreenName' screen will open up.

On the 'Parameters' tab the possible parameters for customization user queries are listed with their type and current value. On the 'Events' tab the customizable events are listed. The 'Exists' checkbox indicates whether there is an existing custom query for the event or not.



2. Customization

2.1. Create custom field

Open the Customization Fields table in SAP Business One via: Tools > User Defined Windows > BXCUSTFD.

Field	Description	Value
Field Name	This name will be used in the custom queries. On the Customization window the field will appear with this name. If the name of an existing field is added, the other properties can be changed. (For example this hide an existing field set the 'Visibility' to 'No')	Please see: Produmex Manufacturing Customization Examples
Field Type	Type of the field.	Supported field types: <i>String(0)</i> : Creates a new input fields. It will have a 'validate' and 'validate after' event. <i>Button(13)</i> : Creates a new button. It will have a 'click' and 'click after' event.
Label	The displayed text. It is also possible the change the label of an existing field.	
Module	The name of the mobile solution.	BXPPSMobilePDC
On External Form	Defines whether the new field is added to a new screen or not. If yes, a new button will appear to open the external screen.	Yes/No
Protected	If set to yes, the field will be displayed on the proceeding screen as well. The entered value in read-only mode.	Yes/No
Read Only	Defines whether the field will be read only or not.	Yes/No
Screen	The screen name.	The screen name can be found in the first line on the Customization window.
Visible	Defines whether the field is displayed on the screen or not.	Yes/No

2.2. Create an SAP user query

Open the Query Manager in SAP Business One via: Tools > Queries > Query Manager

Create the user query.

The name of the query defines when it will be executed, therefore save it as the name of the event when you would like to run the user query.

EXAMPLE: The user query that runs when the 'GR PO' screen is loaded in Produmex Scan is 'BXMobileWH9_GoodsReceiptPOScreen_Load'

2.2.1. Supported message types

The following message types can be used in user queries for Produmex PDC:

- **I: Information**

A green line will appear at the footer of the screen with the message. The event will execute.

- **W: Warning**

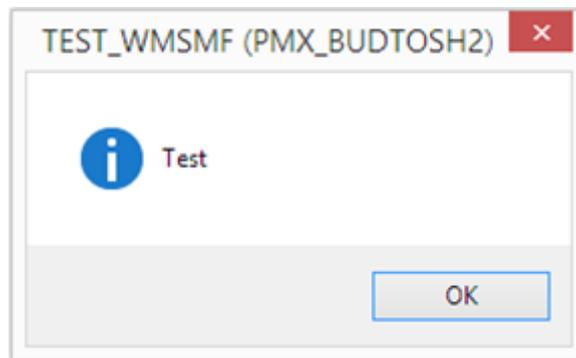
A blue line will appear at the footer of the screen with the message. The event will execute.

- **E: Error**

This is the default message type value. A red line will appear at the footer of the screen with the message. The event not will execute.

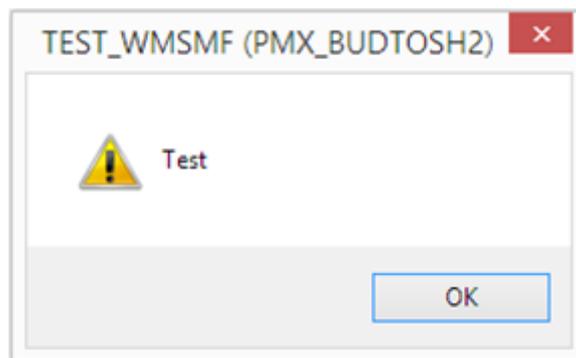
- **IM: Information with message box**

A pop up information message will be prompted, that needs user confirmation (OK). The event will execute.



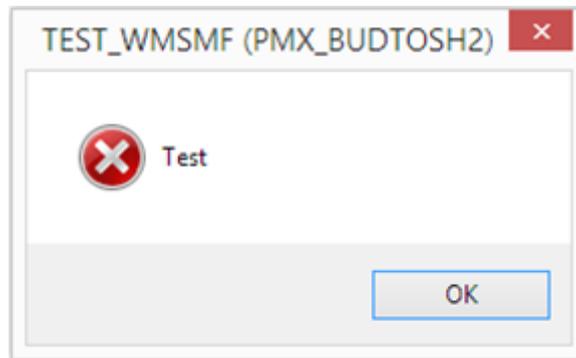
- **WM: Warning with message box**

A pop up warning message will be prompted, that needs user confirmation (OK). The event will execute.



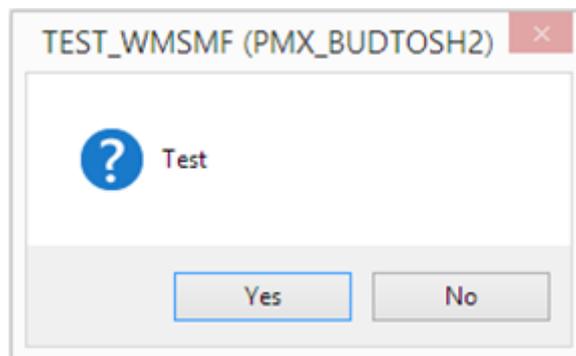
- **EM: Error with message box**

A pop up error message will be prompted, that needs user confirmation (OK). The event will not execute.



- **YM: Yes/No with message box**

A pop up confirmation message will be prompted, that can be answered with yes or no. The event will execute depending on user choice.



3. Restart the application

To apply the customization restart the mobile application.

You must restart the application every time a new user query is created but it is not necessary to restart the application when modifying an existing query.

Customization Examples

Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

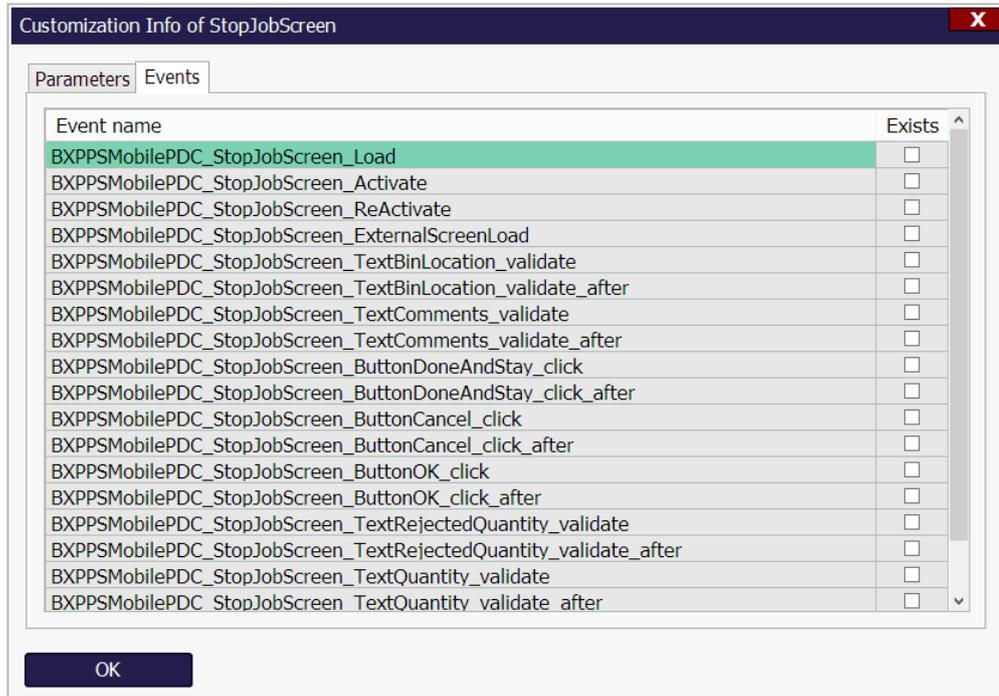
1. Preset field data

With customization it is possible to preset the field data.

Enable the Customization Assist then start Produmex PDC. Go to the screen that you would like to customize and press the Customization Assist button. Because the field data is populated when the screen loads, the event to customize is the load event.

In the example we will preset the 'Quantity' value on the 'Complete Job' screen to the open quantity on the production order.

Go to the Complete Job screen and press the Customization Assist button. The event to customize is the 'BXPPSMobilePDC_StopJobScreen_Load' event.



Create a new query in the Query Manager. User query name will be the name of the event.

In the example the query name will be: 'BXPPSMobilePDC_StopJobScreen_Load'.

Example query:

```
SELECT
CASE WHEN (SELECT SUM(U_BXPCoQty) FROM [@BXPPDCBOOKING] WHERE U_BXPPr00I =
$[CurrentOperation.Code] AND U_BXPIsUnd = 'N') >
$[CurrentOperation.PlannedQuantity]
THEN 0
ELSE $[CurrentOperation.PlannedQuantity] - ISNULL( (SELECT SUM(U_BXPCoQty)
FROM [@BXPPDCBOOKING] WHERE U_BXPPr00I = $[CurrentOperation.Code] AND
U_BXPIsUnd = 'N'), 0)
END AS TextQuantity
```

The query in the example runs when the Complete Job screen loads. It selects the completed quantities from the PDC Booking (@BXPPDCBOOKING) table using the operation code

(\$[StopJobOperation.Code]). Then it subtracts the completed quantities from the Operation Planned Quantity (\$[StopJobOperation.PlannedQuantity]). The query sets the value to zero if the completed quantity is greater than the planned quantity.

After the query has been added, restart Produmex PDC in order to load the customization data.

Please note: *You only need to restart the application once a new query has been added. If you only add modifications to the user query, Produmex PDC doesn't have to be restarted.*

2. Validation

It is possible to add validation for 'validation' and 'button' events with customization.

In the example we will set to show warning message if the user scans an operation on the 'Start Job' screen and there is no PDC booking for the previous operation of the same production order.

Go to the Start Job screen and press the Customization Assist button. For the operation validation we will use the `BXPPSMobilePDC_StartJobScreen_TextOperation_validate` event.



Create a new query in the Query Manager. User query name will be the name of the event.

Query name: **'BXPPSMobilePDC_StartJobScreen_TextOperation_validate'**.

```

DECLARE @DocEntry INT
DECLARE @LineNumber INT
DECLARE @PrevLineNum INT
DECLARE @PrevOperation nvarchar(MAX)
DECLARE @PrevVisOrder nvarchar(MAX)
DECLARE @Warning nvarchar(MAX)
DECLARE @NumberOfBookings nvarchar(MAX)
DECLARE @INPUT nvarchar(MAX)
DECLARE @a INT
SET @Warning = ''
SET @INPUT = $[TextOperation]

SET @DocEntry = 0
SET @LineNumber = 0
SET @a = CHARINDEX('-', @INPUT)
IF @a > 0 BEGIN
    SET @DocEntry = CAST(LEFT(@INPUT, @a - 1) AS INT)
    SET @INPUT = SUBSTRING(@INPUT, @a + 1, LEN(@INPUT) - @a)
    SET @a = CHARINDEX(' ', @INPUT)
    IF @a = 0 BEGIN SET @a = LEN(@INPUT) + 1 END
    SET @LineNumber = CAST(LEFT(@INPUT, @a - 1) AS INT)
END
END

```

```
SELECT TOP 1 @PrevLineNum = WOR1.LineNum, @PrevOperation = WOR1.ItemCode,
@PrevVisOrder = WOR1.VisOrder
FROM WOR1 WHERE WOR1.DocEntry = @DocEntry AND WOR1.U_BXPRowTy = 2
AND WOR1.VisOrder < (SELECT VisOrder FROM WOR1 WHERE DocEntry = @DocEntry
AND LineNum = @LineNumber)
ORDER BY VisOrder DESC
IF @PrevLineNum IS NOT NULL BEGIN
    SELECT @NumberOfBookings = COUNT(*) FROM [@BXPPDCBOOKING] book WHERE
book.U_BXPPrODE = @DocEntry AND book.U_BXPPrODL = @PrevLineNum AND
book.U_BXPIsUnd = 'N'
    IF @NumberOfBookings = 0 BEGIN
        SET @Warning = 'Warning: No PDC booking for previous operation,
continue? Operation:' + @PrevOperation + ', line: ' + CAST(@PrevVisOrder AS
nvarchar(MAX))
    END
END
IF @Warning <> ''
BEGIN
    SELECT @Warning AS 'WarningMessage$', 'W' AS 'MessageType$'
END
```

The example query can return two results:

1. No warning message - if the previous operations have PDC bookings or there is no previous operation.
2. Warning message - if there is no PDC booking for the previous operation.

The user query does the following:

Parses the input (\$[TextOperation]) - the value from the screen field named Operation, separate DocEntry and LineNum. The input will be something like: '10-5', or '10-5 (Otuma...)'

Checks what is the previous operation from the same Production Order.

Checks if the previous operation has a booking (from [@BXPPDCBOOKING] table)

If there is no booking, returns a warning message in the result column named 'WarningMessage\$'

3. Hide existing field

It is possible to hide existing fields with customization.

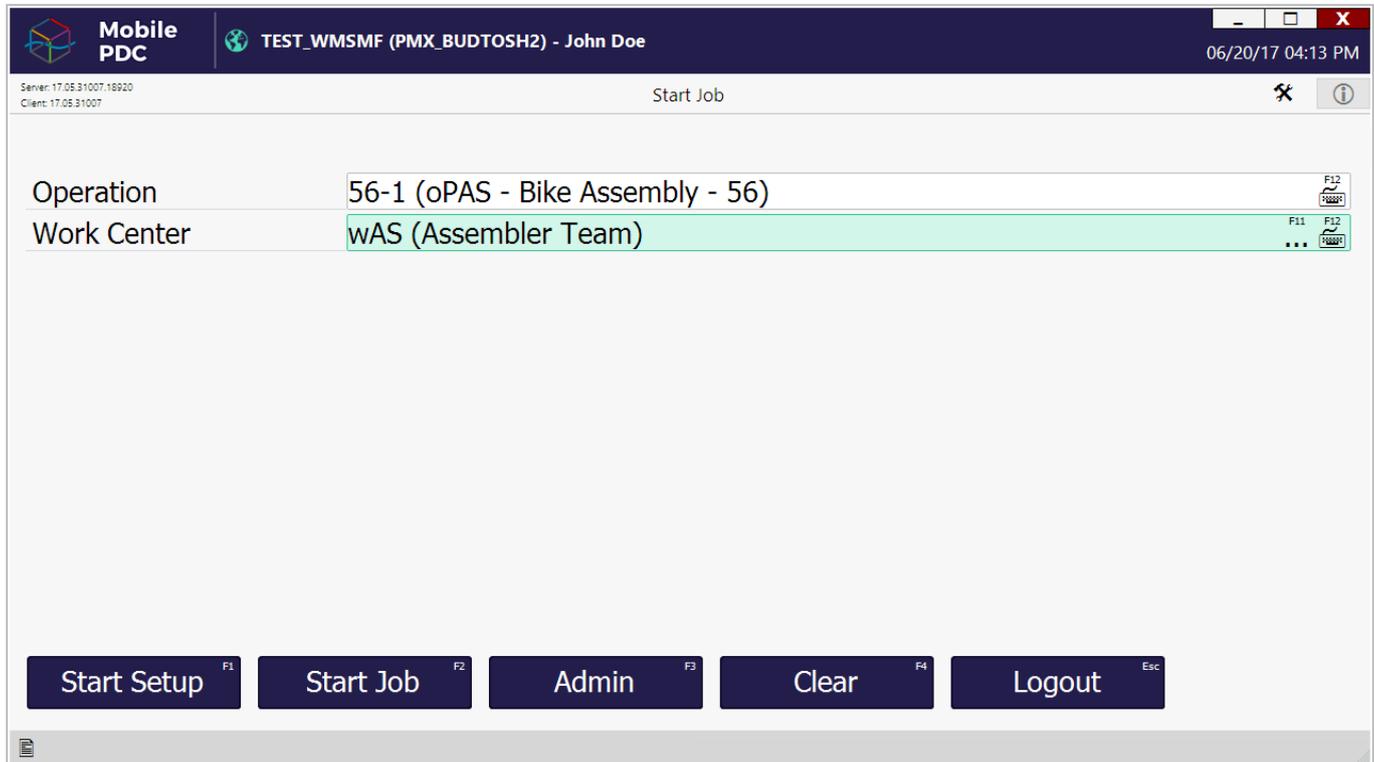
Open the [Customization Fields user table](#) and add a new record for an existing field. Set the *Visible* value to 'No' then press the 'Update' button.

Example: In the example we hide the Default Work Center field from the Start Job screen.

Add the following record to the Customization Fields user table:

Field Name	Module	Screen	Visible
TextDefaultWorkCenter	BXPPSMobilePDC	StartJobScreen	No

Restart Produmex PDC. The customized Start Job screen will look like this:



4. Button customization

It is possible to set an event to trigger a button event.

Example: Automatically press the 'Login' button after the employee code has been added on the LOGIN Screen.

Query name: *BXPPSMobilePDC_LoginScreen_TextEmployeeID_validate_after*

```
IF $[TextEmployeeID] <> ''
BEGIN
SELECT 'ButtonLogin' AS 'Click$'
END
```

5. Custom message

It is possible to add custom messages to the events.

Example: Display a warning message if the Employee ID is not added before pressing the 'Login' button.

Query name: *BXPPSMobilePDC_LoginScreen_ButtonLogin_click*

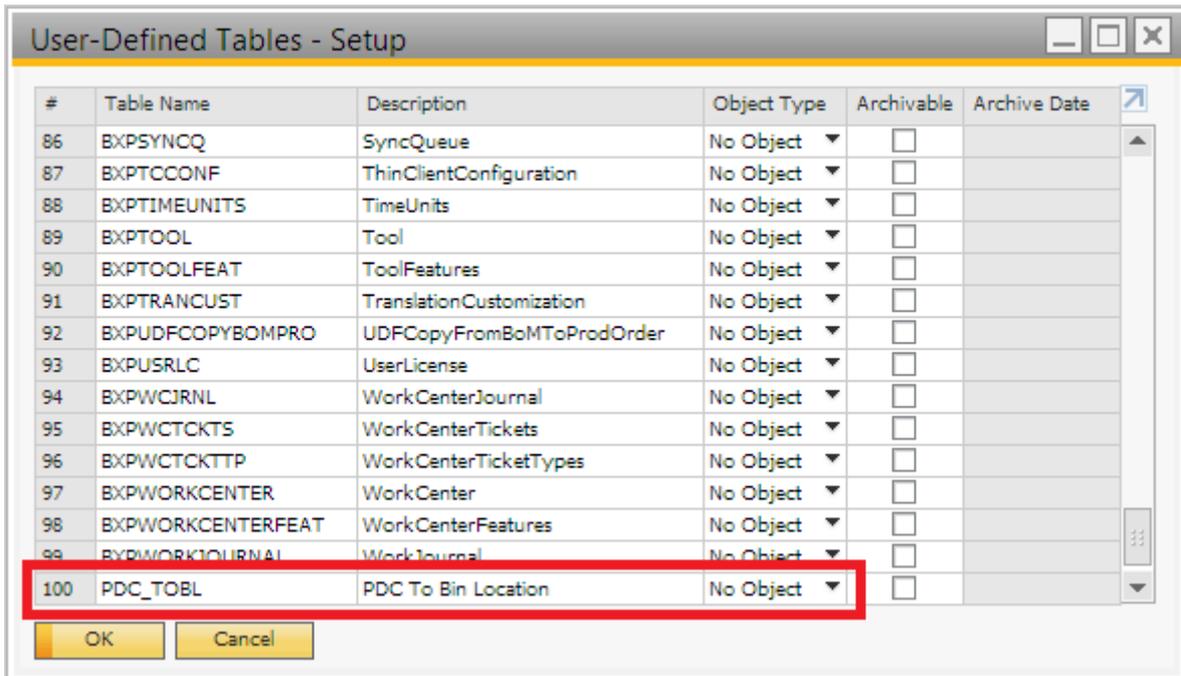
```
IF $[TextEmployeeID] = ''
SELECT 'Scan Employee ID' AS 'Message$', 'E' AS 'MessageType$'
```

For more information about the supported message types please see: [Supported message types](#)

6. Save the Destination Bin Location for the employee

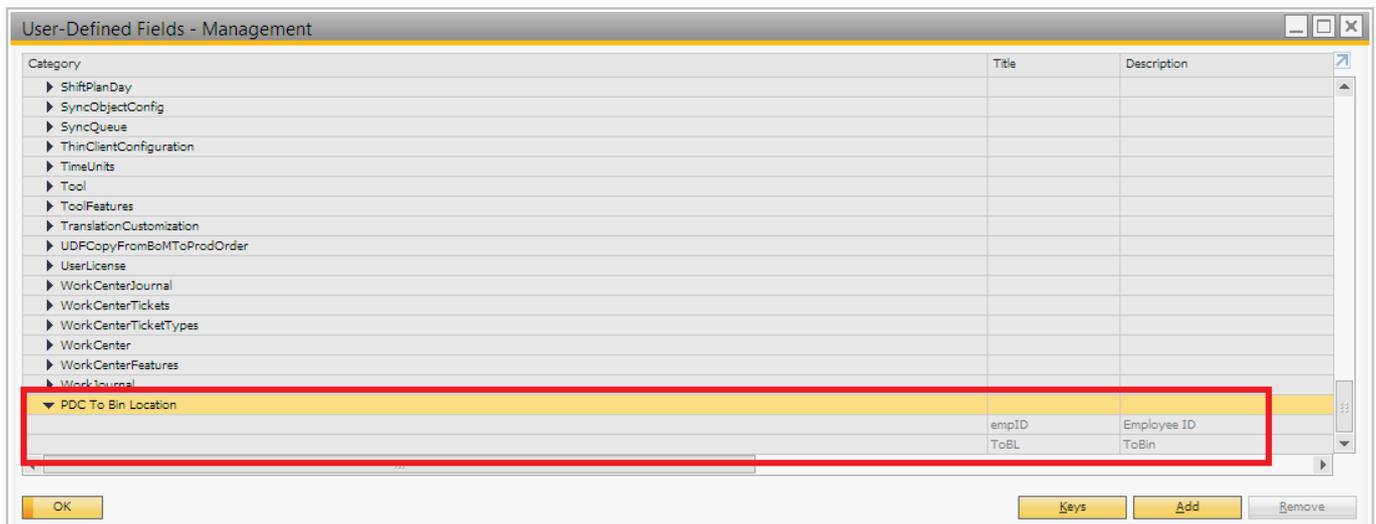
This customization will save the last added bin location on the Products screen. When this screen next loads to the employee, the Bin Location code will be automatically filled. This way if the employee uses the same destination bin location, (s)he does not have to re-enter it.

Create a new user table in SAP Business One. (In the example: PDC_TOBL). Register the new user table as an object.

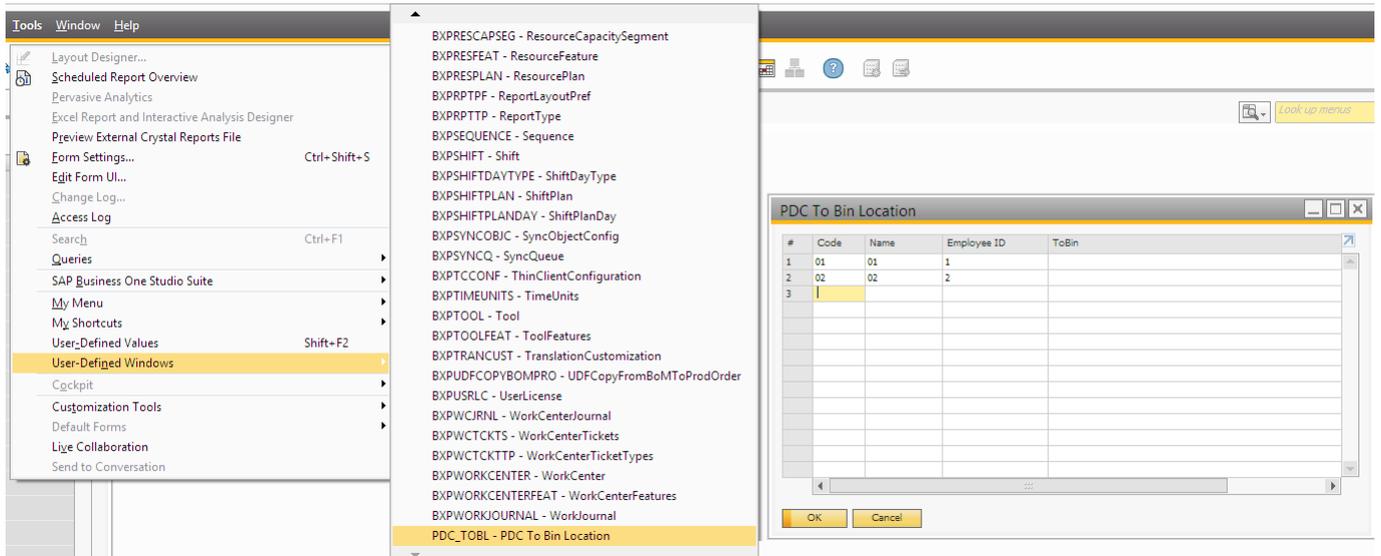


Add the following fields to the user table: [Tools>Customization Tools>User Defined Fields-Management]

- Employee ID (empID)
- To Bin Location (ToBL)



Define the employees on the user table.



Add the following user queries:

6.1. Save the destination Bin Location to the employee

This query will save the added destination Bin Location to the U_ToBL field.

Query name: *BXPPSMobilePDC_ProductsAdvScreen_TextBinLocation_validate_after*

Example query:

```
IF (SELECT ISNULL ([@PDC_TOBL].U_ToBL, '' )
FROM [@PDC_TOBL]
WHERE U_empID = $('[Employee.EmpNumber]') <> $('[TextBinLocation]')
UPDATE [@PDC_TOBL] SET [U_ToBL]= $('[TextBinLocation]') WHERE U_empID =
 $('[Employee.EmpNumber]')
```

6.2. Load the saved Bin Location

This query will populate the Bin Location field with the saved value when the screen loads.

Query name: *BXPPSMobilePDC_ProductsAdvScreen_Load*

Example query:

```
SELECT
(SELECT [@PDC_TOBL].U_ToBL FROM [@PDC_TOBL] WHERE U_empID =
 $('[Employee.EmpNumber]')
AS 'TextBinLocation'
```

How to customize the Produmex Manufacturing Reports in MSSQL

Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

Produmex Manufacturing offers an easy solution for report customization with Crystal Reports.
Please note: This document describes the custom report creation procedure in MSSQL.

In our example we will include the foreign name of the operation on the Job Requirements report under the operation name.

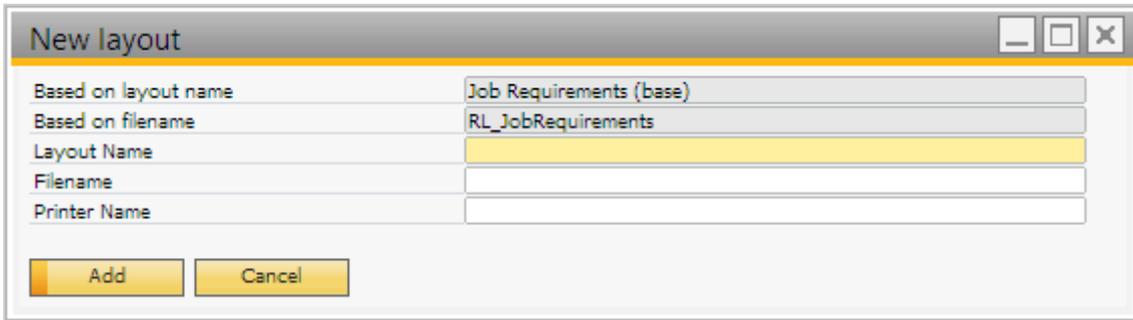


Open the report. On the 'Select Report Layout' form select a base layout on the grid then click on the 'New' button to create a new layout.

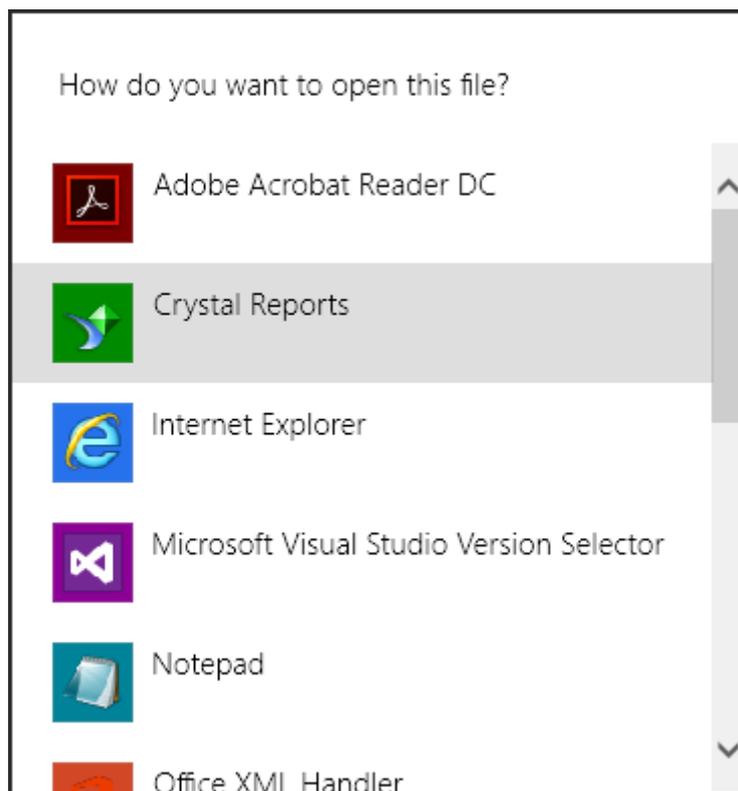
In the example we will open the Job Requirements Report.

Report Layout	File	Printer Name
Job Requirements (base) (default)	RL_JobRequirements	

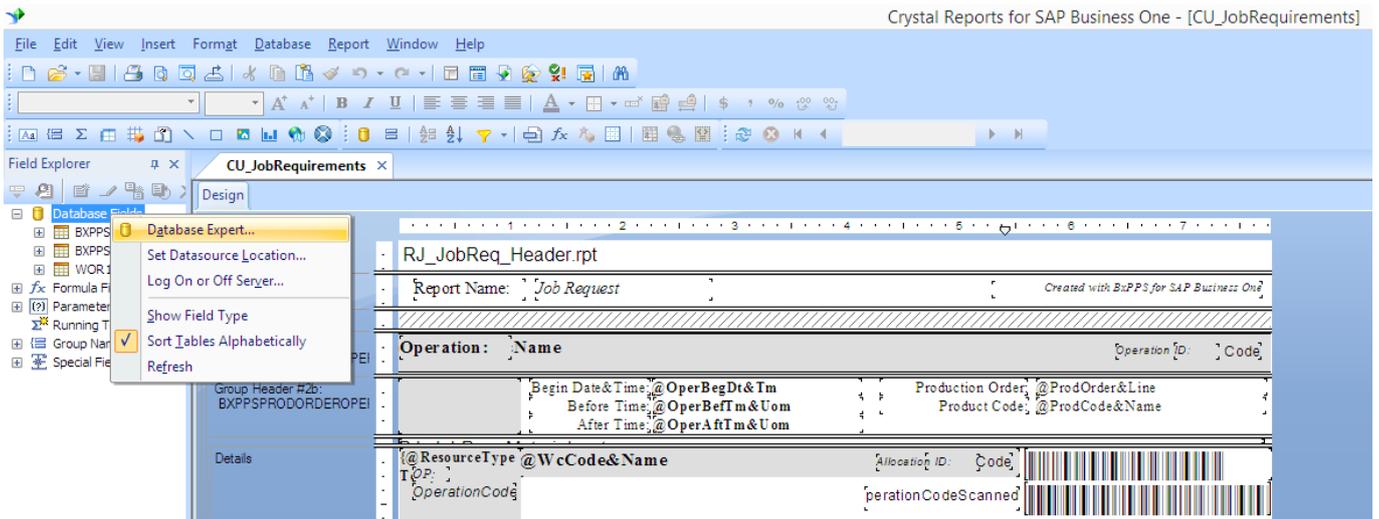
Add the 'Layout Name' and the 'Filename' then click on the 'Add' button. The base layout will be copied to the new layout.



After adding the custom layout, select it on the grid. Click on the 'Edit Layout' button and open the layout with Crystal Reports.



After the layout has been opened in Crystal Reports, setup the ODBC connection between the report and the database. On the Field Explorer right click on 'Database Fields' and select 'Database Expert' option.



On the opening 'Database Expert' form open the 'Create New Connection' folder and click on the 'OLE DB' folder. On the OLE DB form chose 'Microsoft OLE DB Provider for SQL Server' as the Provider. Click on 'Next' to proceed.

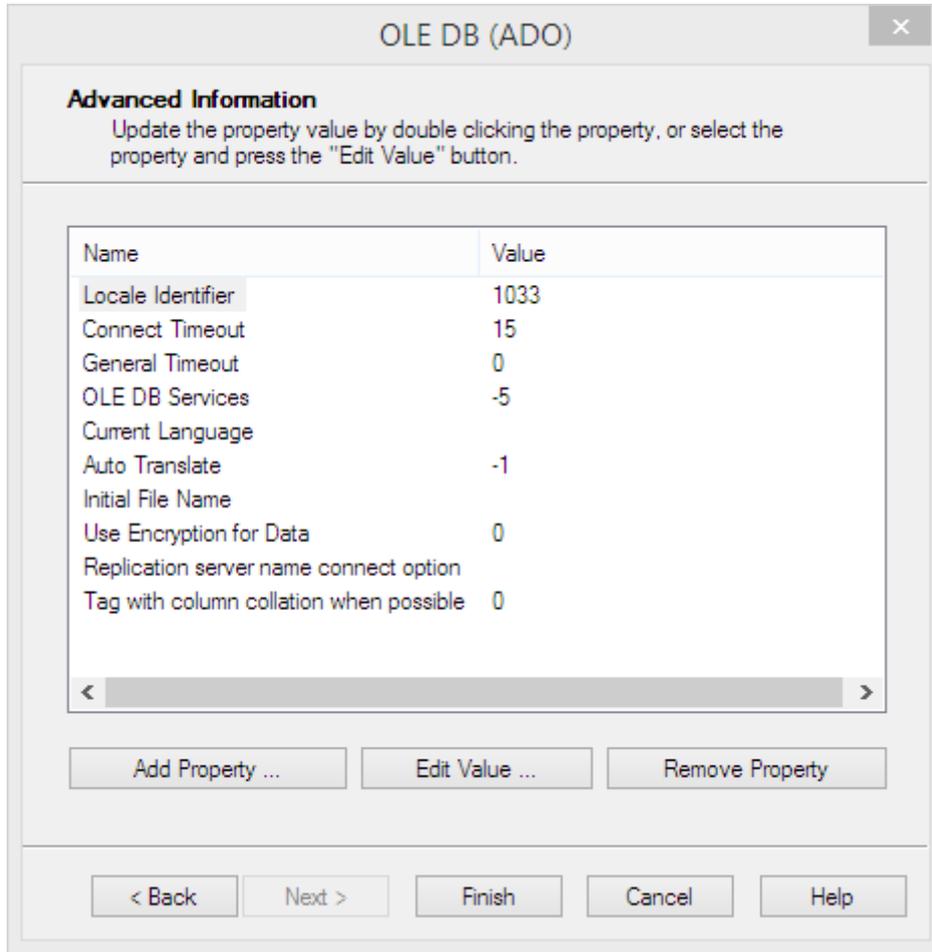


On the next screen add the connection information. Select the local server as the 'Server' and enter the User ID and Password. Then select the database from the dropdown menu and click on 'Next'.

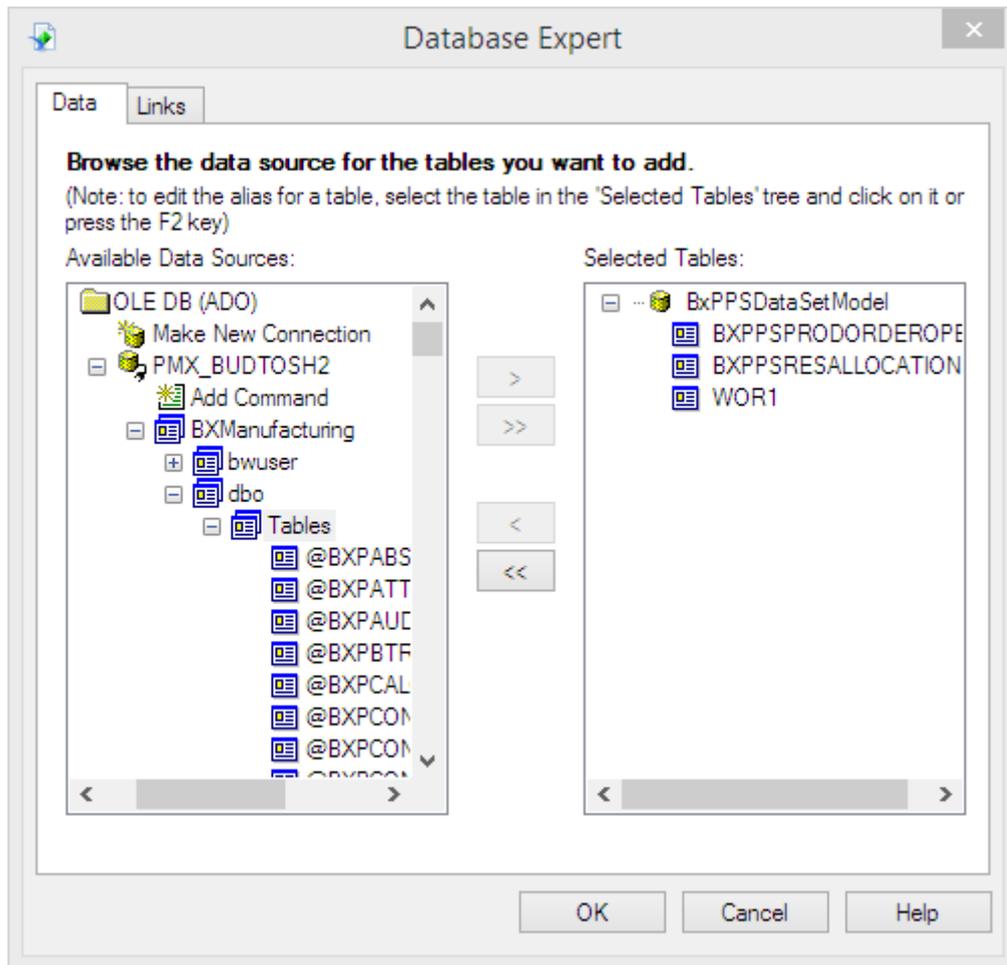
Please note: It is possible the use the custom report in a different company as well, because the Produmex Manufacturing add-on will always set the Server and the Database name to the currently connected Company database.



On the next screen click on 'Finish'.

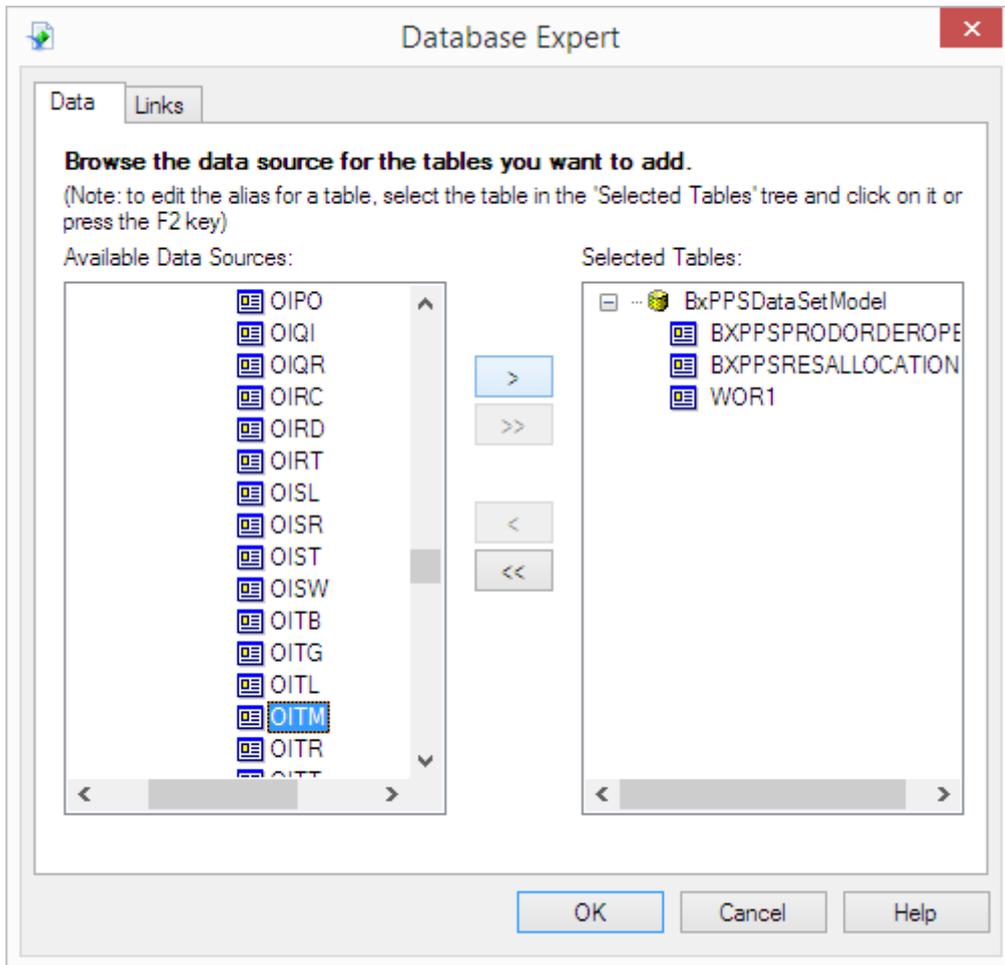


The database is added to the OLE DB folder.



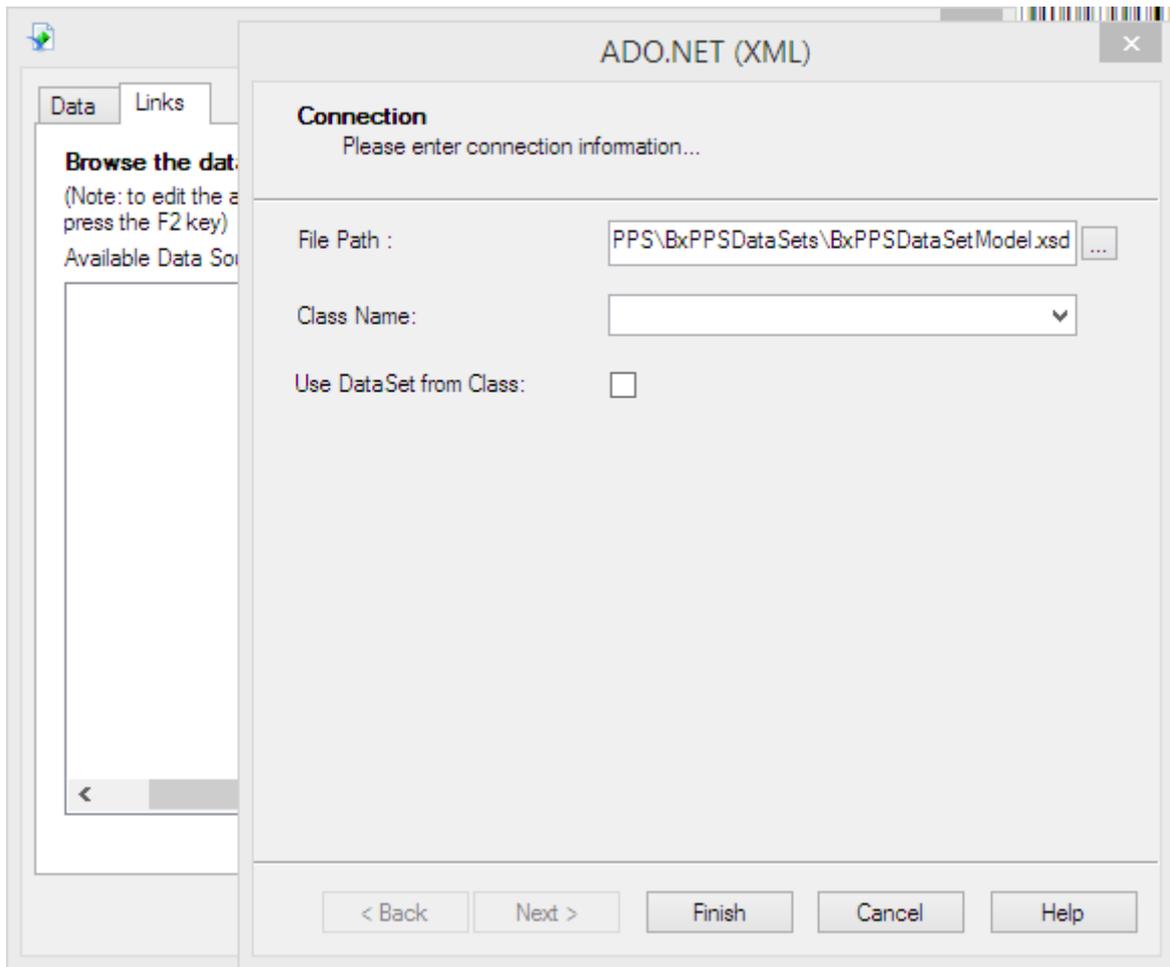
Then select the database table that contains the field you would like to add and click on the right arrow to add it to the 'Selected Tables'.

In our example we would like to add the foreign name of the operation to the report. In order to do this, we have to add the OITM table (Item Master Data).

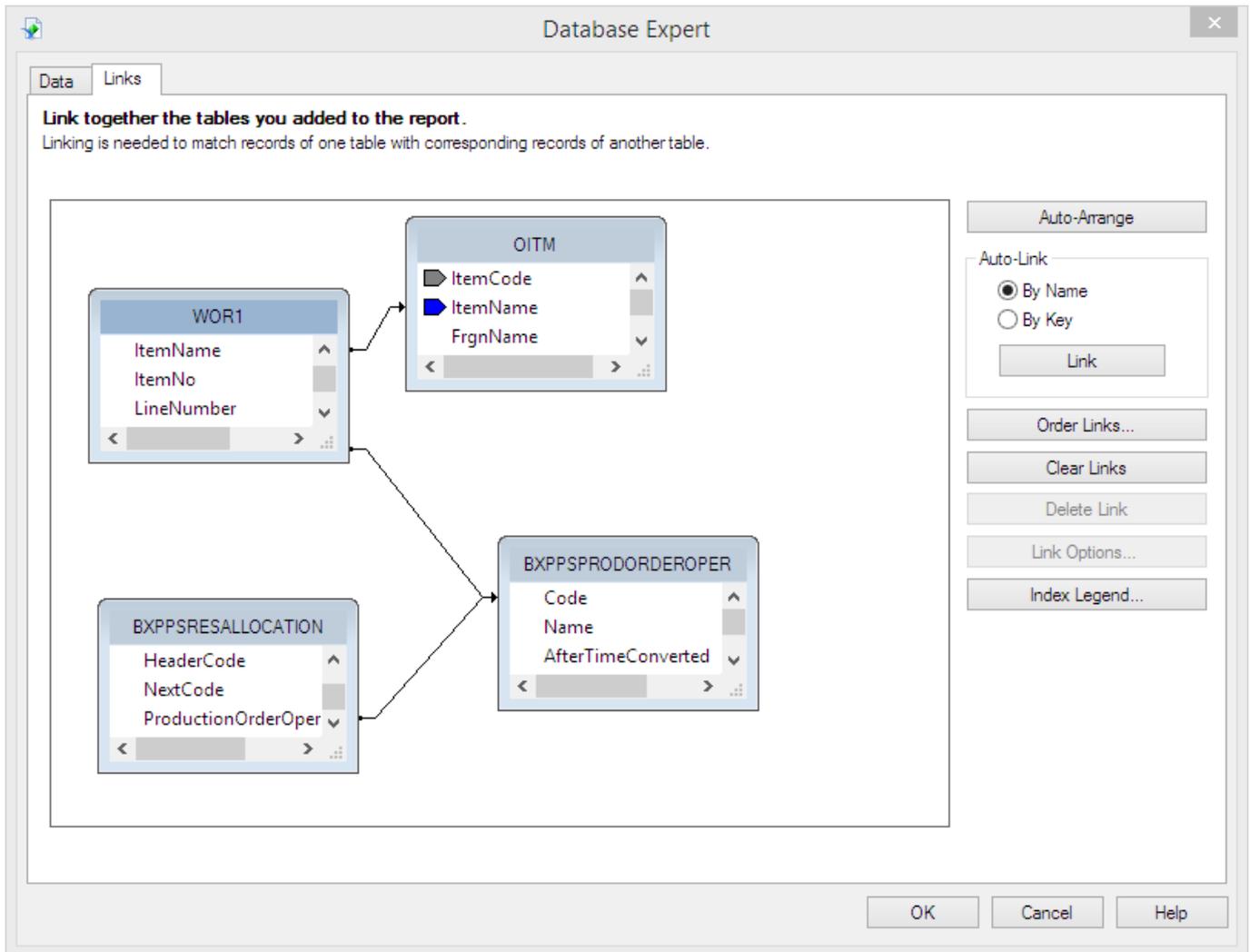


Define the file path to the BX PPS Data Set Model by clicking on the 'Links' tab. Browse the BxPPSDataSetModel.xsd file after clicking on the '...' button on the 'ADO.NET form. Then click on 'Finish'.

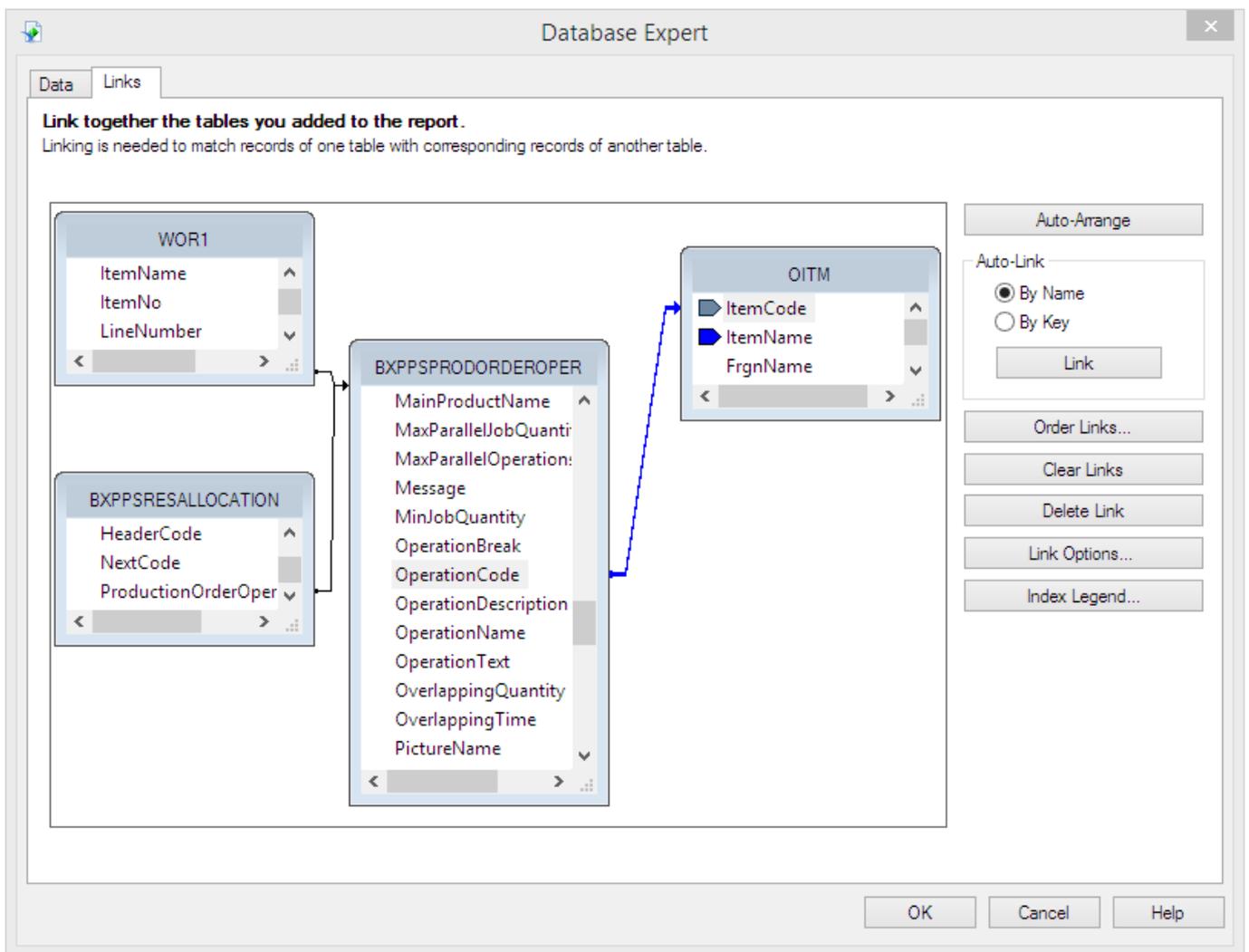
The default file path is: *C:\Program Files\SAP\SAP Business One\AddOns\BXP\Produmex Manufacturing\BxPPSDataSetModel.xsd*



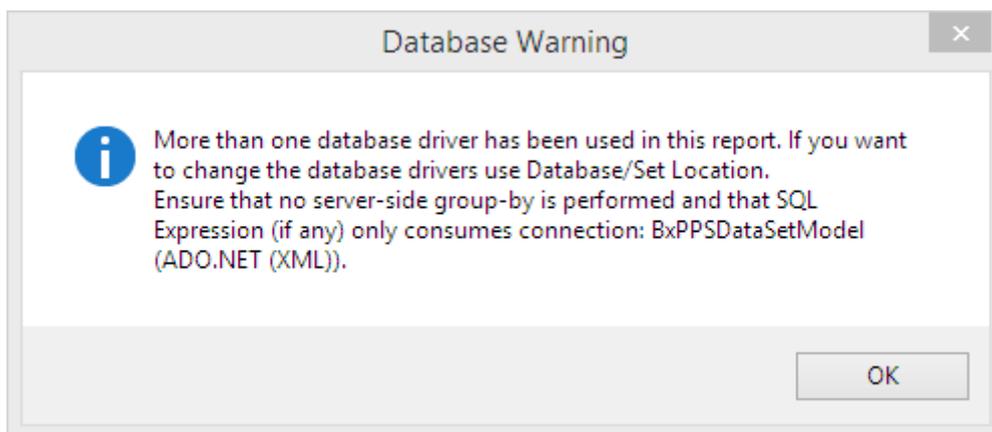
On the Links tab the links between the table records are shown. These links are created automatically and might not be correct for the customization goals. Correct the links by deleting the wrong ones and creating new links. Click on the 'OK' button.



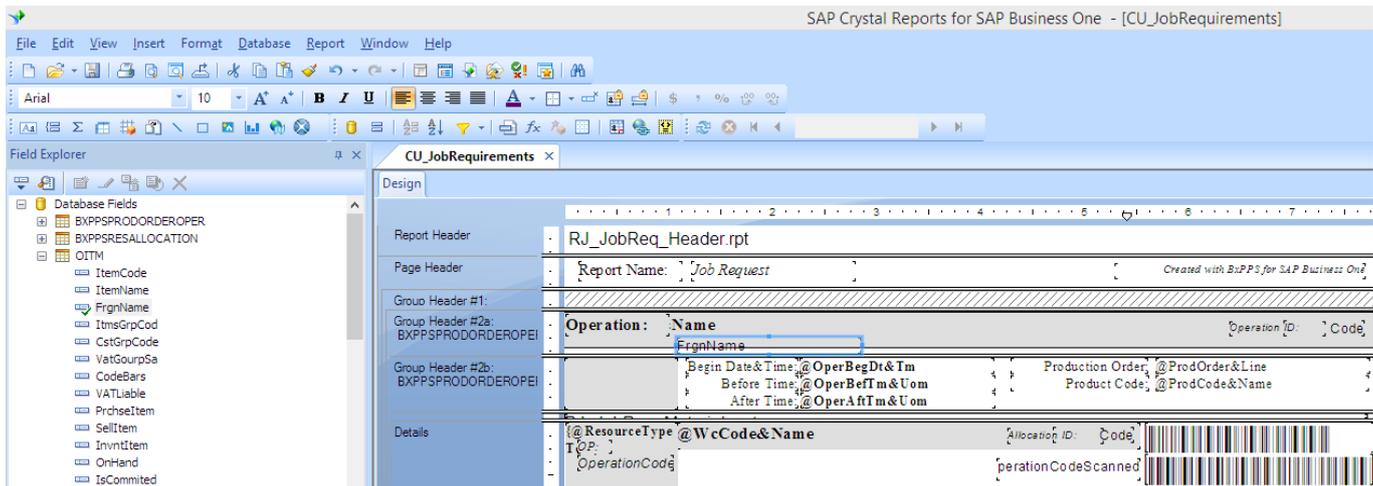
In the example the 'Item Name' field from the OITM table was linked automatically to the WOR1 table. First we delete this link then we create the new one by connecting the Operation Code from the BXPPSPRODORDEROPER table to the Item Code on the OITM table.



A database warning will pop up. Click on the 'OK' button to acknowledge the message.



Select the field to add to the report on the 'Field Explorer' module and simply drag and drop it to the report.



After the formatting is ready, save the report. Make sure that you do not save it under *SAP\SAP Business One\AddOns* because in that case the file will be overwritten by a new Produmex add-on installation. It is also recommended to save the custom report to a central location where every client will be able to see it.

When you print the custom report created in the example, the foreign name will be added under the operation name.



Special customization

Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

1. User queries for validation

The user queries for validation should return two columns:

- RowTy: row type can be E (error) or W (warning)
- Msg: the error/warning description

An error message will appear in red on the bottom of the screen, and block the current process. A warning message will be displayed in a separate window, asking a confirmation from the user to continue. For more information about the message types please see: [Supported message types](#)

Only the first row of the result is taken into account.

1.1. User identification

User query name: *bxtc_pdc_user_identification*

Parameters:

- [%0]: Action code (1 = OK)=/
- [%1]: Employee ID (OHEM.empID)
- [%9]: Terminal ID (IP address)

1.2. Production Operation Selection

User query name: *bxtc_pdc_production_operation_selection*

Parameters:

- [%0]: Action code ; possible values are:
 - 1 = Complete/Stop
 - 2 = Break/Partial
 - 3 = Resume (only available in old web/tomcat PDC client)
 - 4 = New Op
 - 5 = Admin
 - 6 = Materials (not available yet in BX Mobile PDC client)
- [%1]: Employee ID (OHEM.empID)
- [%2]: Operation ID (WOR1.U_BXPBxID)
- [%3]: Job/Setup (0 = Job, 1 = Setup)
- [%9]: Terminal ID (IP address)

If more than one operations are selected, then parameters [%2] and [%3] are a coma-separated list of values.

1.3. Completed/Partial Job Quantities

User query name: *bxtc_pdc_job_quantities*

Parameters:

- [%0]: Action code (1 = OK)
- [%1]: Employee ID (OHEM.empID)
- [%2]: Operation ID (WOR1.U_BXPBxID)
- [%3]: Completed Quantity
- [%4]: Rejected Quantity
- [%5]: PDC Booking Code
- [%6]: Duration (calculated) in seconds (not supported yet)
- [%7]: Work Center Code
- [%9]: Terminal ID (IP address)

1.4. Start New Operation

User query name: *bxtc_pdc_start_operation*

Parameters:

- [%0]: Action code (1 = OK)
- [%1]: Employee ID (OHEM.empID)
- [%2]: Operation ID (WOR1.U_BXPBxID)
- [%3]: Job/Setup (0 = Job, 1 = Setup)
- [%9]: Terminal ID (IP address)

2. Other user queries

2.1. Product serial/batch number

This query is used when the serial/batch selection is disabled for products by enabling the ‘Skip product serial/batch quantities screen’ option on the [Thin client 2](#) tab. In that case, the operation quantities can be entered without selecting batches/serial numbers, and the user query will automatically create them when the PDC booking is being processed.

User query name: *bxtc_pdc_serial_batch_products*

Parameters	Returned columns
[%1]: Employee ID (OHEM.empID) [%2]: Operation ID (WOR1.U_BXPBxID) [%3]: PDC Booking Code [%4]: Product Code (OITM.ItemCode) [%5]: Quantity [%6]: Is Rejected? (0 = Completed, 1 = Rejected) [%7]: Purchase Order Line Extension Code (outsourcing only) [%8]: Product Type (outsourcing only) (1 = Main Product, 2 = By-Product, 3 = Unfinished Product)	SBNum: Serial or batch number Qty: Quantity

2.2. Material serial/batch number

This query is used when the serial/batch selection is disabled for materials by enabling the ‘Skip material serial/batch quantities screen’ option on the [Thin client 2](#) tab. In that case, the operation quantities can be entered without selecting batches/serial numbers, and the user query will automatically create them when the PDC booking is being processed.

User query name: *bxtc_pdc_serial_batch_materials*

Parameters	Returned columns
[%1]: Employee ID (OHEM.empID) [%2]: Operation ID (WOR1.U_BXPBxID) [%3]: PDC Booking Code [%4]: Material Code (OITM.ItemCode) [%5]: Quantity [%7]: Purchase Order Line Extension Code	SBNum: Serial or batch number Qty: Quantity

2.3. PTM Log

User query name: *bxtc_pdc_ptm_log_query*

Example query:

```
EXECUTE sp_executesql N'
BEGIN

DECLARE @RowNum INT, @IsStarted VARCHAR(1), @StartDate VARCHAR(20)

SET NOCOUNT ON

CREATE TABLE #EmpLog (RowNum INT, IsStarted VARCHAR(1), StartDate
VARCHAR(50), EndDate VARCHAR(50), Employee INT)
INSERT INTO #EmpLog SELECT ROW_NUMBER() OVER(ORDER BY U_BXPStam0) AS RowNum,
U_BXPActn AS IsStarted, U_BXPStam0 AS StartDate, NULL AS EndDate, U_BXPEmpID
AS Employee
FROM [dbo].[@BXPATTLOG]
WHERE U_BXPEmpID = 1

DECLARE log_cursor CURSOR FOR
SELECT RowNum, IsStarted, StartDate
FROM #EmpLog

OPEN log_cursor
FETCH NEXT FROM log_cursor
INTO @RowNum, @IsStarted, @StartDate;
WHILE @@FETCH_STATUS = 0
BEGIN
    IF @IsStarted = ''N''
    BEGIN
        UPDATE #EmpLog SET EndDate = @StartDate
        WHERE RowNum = @RowNum - 1
    END
    FETCH NEXT FROM log_cursor
    INTO @RowNum, @IsStarted, @StartDate;
END
CLOSE log_cursor;
DEALLOCATE log_cursor;

SELECT
CAST(LEFT(StartDate, 8) AS DATE) AS Date,
CAST(LEFT(StartDate, 8) AS DATE) AS StartDate,
CAST(LEFT(RIGHT(StartDate, 6), 4) AS INT) AS StartTime,
CAST(LEFT(EndDate, 8) AS DATE) AS EndDate,
CAST(LEFT(RIGHT(EndDate, 6), 4) AS INT) AS EndTime,
NULL AS BreakTime,
NULL AS JobTime,
```

```

DATEDIFF(MINUTE,
CAST(LEFT(StartDate, 8) AS DATETIME)
+ CAST(DATEADD(HOUR, (CAST(SUBSTRING(StartDate, 9, 2) AS INT)),
DATEADD(MINUTE, (CAST(SUBSTRING(StartDate, 11, 2) AS INT)),
DATEADD(SECOND, CAST(SUBSTRING(StartDate, 13, 2) AS INT), CAST(''00:00:00''
AS TIME(3)))))) AS DATETIME),
CAST(LEFT(EndDate, 8) AS DATETIME)
+ CAST(DATEADD(HOUR, (CAST(SUBSTRING(EndDate, 9, 2) AS INT)),
DATEADD(MINUTE, (CAST(SUBSTRING(EndDate, 11, 2) AS INT)),
DATEADD(SECOND, CAST(SUBSTRING(EndDate, 13, 2) AS INT), CAST(''00:00:00'' AS
TIME(3)))))) AS DATETIME)) AS WorkTime,

CASE WHEN DATENAME(dw, CAST(LEFT(StartDate, 8) AS DATETIME)) IN
(''Saturday'', ''Sunday'') OR EXISTS (SELECT 1 FROM HLD1 WHERE StrDate =
CAST(LEFT(StartDate, 8) AS DATETIME)) THEN ''N'' ELSE ''Y'' END AS
IsWorkingDay,

NULL AS InfoText1,
NULL AS InfoText2

FROM #EmpLog WHERE IsStarted = ''Y'' AND StartDate IS NOT NULL AND EndDate
IS NOT NULL

DROP TABLE #EmpLog
END '

```

2.4. Workshop Monitor

User query name: *bxtc_pdc_workshop_monitor_query*

Example query:

```

SELECT TOP 30
    T0.Code as Code,
    T0.Name as Name,
    T1.ItemCode as ProductCode,
    T3.ItemName as ProductName,
    T0.U_BXPOpCod as OperationCode,
    T0.U_BXPOpNam as OperationName,
    T5.U_BXPPrfWC as PreferredWorkCenter,
    T5.U_BXPFeat as PreferredFeature,
    T0.U_BXPPlQty as PlannedQuantity,
    T0.U_BXPCoQty as CompletedQuantity,
    T0.U_BXPRejQt as RejectedQuantity,
    T2.IssuedQty AS IssuedQuantity,
    T1.DocEntry as DocEntry,
    T1.DocNum as DocNum,
    T2.LineNum as LineNum,
    T0.U_BXPBSetu as SetupTime,
    T0.U_BXPEDuDt as DueDate,

```

```
T0.U_BXPPDueT as DueTime,
T4.U_BXPPstCd AS PDCPostingCode,
T4.U_BXPPstDt as PDCPostingDate,
T4.U_BXPPstTm as PDCPostingTime,
T4.U_BXPWCent as PDCWorkCenter,
T4.Code as PDCBookingID,
T6.firstName as EmployeeFirstName,
T6.lastName as EmployeeLastName,
T6.empID as EmployeeID

FROM
    [@BXPPRODORDEROPER] T0
    INNER JOIN [OWOR] T1 ON T0.U_BXPPrODE = T1.DocEntry AND
T1.[Status] <> N'L'
    INNER JOIN WOR1 T2 ON T2.U_BXPBxID = T0.Code
    INNER JOIN OITM T3 ON T3.ItemCode = T1.ItemCode
    LEFT OUTER JOIN [@BXPPDCBOOKING] T4 ON T4.U_BXPPr00I =
T0.Code AND T4.Code IN
    (SELECT TX.Code FROM [@BXPPDCBOOKING] TX WHERE
TX.U_BXPIsUnd = 'N' AND TX.U_BXPPr00I = T0.Code AND
    TX.U_BXPPstDt = (SELECT MAX(U_BXPPstDt) FROM
[@BXPPDCBOOKING] TX1 WHERE TX1.U_BXPIsUnd = 'N' AND TX1.U_BXPPr00I =
T0.Code) AND
    TX.U_BXPPstTm = (SELECT MAX(U_BXPPstTm) FROM
[@BXPPDCBOOKING] TX2 WHERE TX2.U_BXPIsUnd = 'N' AND TX2.U_BXPPr00I = T0.Code
AND TX2.U_BXPPstDt = TX.U_BXPPstDt))
    LEFT OUTER JOIN [@BXPPRODORDERREQU] T5 ON T5.U_BXPPr00I =
T0.Code AND T5.U_BXPResTy = 1
    LEFT OUTER JOIN OHEM T6 ON T6.empID = T4.U_BXPEmpID
WHERE
    T0.U_BXPPlQty > T0.U_BXPCoQty + T0.U_BXPREjQt AND
T0.U_BXPCoQty + T0.U_BXPREjQt > 0 AND
T0.U_BXPIsOuS = 'N'

ORDER BY
    T0.U_BXPEndDt DESC,
    T0.U_BXPEndTm DESC
```

When using the example query, operations with completed jobs are listed where the booked quantity is less than the planned quantity and there is at least one booking on it. Outsourced operations are not displayed.

2.5. Personal duration factor for multiple PDC bookings

This query is used when reporting a completion (either partial or complete) for more than one running jobs at the same time. In that case, the machine duration will be the full duration for all PDC bookings, however the person duration will be split among the PDC bookings according to the logic defined in this user query.

User query name: *bxtc_pdc_multiple_pdc_person_duration_factor*

Parameters	Returned columns
[%1]: Employee ID (OHEM.empID) [%2]: List of operation codes, coma-separated [%3]: List of work center codes, coma-separated (in the same order) [%4]: List of completed quantities, coma-separated (in the same order) [%5]: List of rejected quantities, coma-separated (in the same order) [%9]: Terminal ID (IP address)	OpCod: operation code PsDur: person duration

Please note: This query has a default implementation which divides the total duration by the number of PDC bookings processed together for each PDC booking.

2.6. Order Recommendation Custom Grouping

This query is used to customize the auto grouping function for [MTO planning](#).

The query runs when:

- If you click on Auto-Group button on the [Group Recommendations form](#)
- Order recommendations are being created and the 'Auto Group' option is set to true on the [MTO tab](#) of Produmex Manufacturing settings

User query name: *BXPPS_MTO_QueryNonGroupableItems*

Parameters:

- [%0]: MTO scenario code

Example query:

```
SELECT Code FROM [@BXPMT0ORDRSOLREF]
WHERE U_BXPMT0Sc = '[%0]' AND U_BXPitmCd IN (
    SELECT OITM.ItemCode
    FROM OITM, itt1
    WHERE oitm.itemcode = itt1.father and (itt1.code='ITEM01' or
    itt1.code='ITEM02' or itt1.code='ITEM03') )
```

Set up Password Protected Login for Produmex PDC

Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

Add a new PDC Password user field to the Employee Master Data table.



Set the password on this field.



Open Produmex Manufacturing settings. Set the new user field as the 'Employee Card Code Field' on the PDC tab.



Enable the 'Login is Password protected' setting on the Thin Client 2 tab.



In order to login on the terminal, the employee has to enter the password instead of the employee number.



How to connect a scale to Produmex PDC

Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

This customization example shows how to use a scale in Produmex PDC with the help of the Produmex WMS ScaleComm Service.

We will add a new Weigh button to the screens where the products can be received and materials can be issued. When this button is pressed, a user query fills the product/material Quantity field with the measured weight. The used scale will be identified based on the work center.

Prerequisites:

- Installed Produmex Manufacturing and Produmex PDC
- Installed Produmex WMS up to version 7.0
- Installed Produmex ScaleCommService

For more information about Produmex PDC customization please see: [Customization Technology](#).

1. Add Weigh button to PDC screens

Open the Customization Fields table via: Tools > User Defined Windows > BXPCUSTFD - Customization Fields. For more information about the Customization Fields user table please see: [Create custom field](#)

Add the Weigh button to the following screens:

- Stop Job/Partial Job
- Product (for by-products)
- Product batch

- Material
- Material batch

Field Name	Field Type	Label	Screen	Visible	Protected
PWeigh	Button	Weigh	StopJobScreen	Yes	No
BPWeigh	Button	Weigh	ProductsAdvScreen	Yes	No
MWeigh	Button	Weigh	MaterialsAdvScreen	Yes	No
PBWeigh	Button	Weigh	ProductBatchNumbersAdvScreen	Yes	No
MBWeigh	Button	Weigh	MaterialBatchNumberPickerAdvScreen	Yes	No

In order to use the UoM conversion for batch managed items, add the following user fields on the Customization Fields table as well.

Field Name	Field Type	Screen	Visible	Protected	ReadOnly
ItemM	String	MaterialsAdvScreen	No	Yes	Yes
ItemP	String	ProductsAdvScreen	No	Yes	Yes

2. Add user fields

The used scale will be identified based on the work center. In order to assign a work center to a scale, add a new user field to the [Scale Weigh result user table](#).

In order to add UoM conversion, define a UoM for PDC user field as well.

Open the User-Defined Fields- Management screen from Tools > Customization Tools and add the work center field and PDC UoM field.

In the example we added the following user fields:

Title	Description	Type	Length
WORKCENTER	Work center	Alphanumeric	50
UOMPDC	UoM for PDC	Alphanumeric	2

Register the fields with the Objects Registration Wizard. Make sure to mark the new fields as 'Visible' during the registration. *Please note: A work center can only have one assigned scale.*

In order to convert between units of measurements correctly, make sure that both the UoM of the item and the UoM of the scale is defined properly on the following standard SBO tables:

- Weight - Setup (OWGT)
- Units of Measure - Setup (OUOM)

Please note: When using SAP HANA, value names are case sensitive.

3.4. PDC Settings

Make sure that the 'PDC Custom Validation UQ setting enabled' option is enabled on the [Thin Client tab](#) of Produmex Manufacturing Settings.

4. Add the custom queries

Add the custom queries with the Query Manager. For more information about PDC custom queries please see: [Create a SAP user query](#).

4.1. Product

Query name: *BXPPSMobilePDC_StopJobScreen_PWeigh_click*

```
SELECT
'TextQuantity' as Click$,
(
SELECT [@PMX_SCWR].U_PMX_WEIGH * OWGT.WightInMG / OWGT2.WightInMG
FROM [@PMX_SCWR]
JOIN OWGT ON OWGT."UnitDisply" = [@PMX_SCWR].U_UOMPDC
JOIN OITM ON OITM."ItemCode" = $[CurrentPDCBooking.ProductCode]
JOIN OWGT AS OWGT2 ON OWGT2."UnitDisply" = OITM."InvntryUom"
WHERE U_WORKCENTER= $[CurrentPDCBooking.WorkCenterCode]
)
AS TextQuantity,
'TextRejectedQuantity' as Click$
```

Query name: *BXPPSMobilePDC_StopJobScreen_PWeigh_click_after*

```
SELECT 'TextRejectedQuantity ' as Click$
```

4.2. By-product

Query name: *BXPPSMobilePDC_ProductsAdvScreen_BPWeigh_click*

```
SELECT
'TextQuantity' as Click$,
(
SELECT [@PMX_SCWR].U_PMX_WEIGH * OWGT.WightInMG / OWGT2.WightInMG
FROM [@PMX_SCWR]
JOIN OWGT ON OWGT."UnitDisply" = [@PMX_SCWR].U_UOMPDC
JOIN OITM ON OITM."ItemCode" = $[SelectedProduct.ProductCode]
JOIN OWGT AS OWGT2 ON OWGT2."UnitDisply" = OITM."InvntryUom"
WHERE U_WORKCENTER= $[CurrentPDCBooking.WorkCenterCode]
)
AS TextQuantity
```

This query runs when the user clicks on the 'Weigh' button.

First it clicks in the Quantity field.

Then it selects the weigh from the PMX_SCWR table based on the current work center and it converts the measured weigh from the UoM of the scale defined on the UoM for PDC field to the UoM of the item specified on the Produmex Inventory tab of the Item Master Data.

Query name: *BXPPSMobilePDC_ProductsAdvScreen_BPWeigh_click_after*

```
SELECT 'TextProduct' as Click$
```

This query runs after the click on the 'Weigh' button. It clicks in the Product field in order to validate the quantity added with the previous query.

4.3. Product and by-product batch

Query name: *BXPPSMobilePDC_ProductsAdvScreen_ButtonSerialBatch_click*

```
SELECT $[SelectedProduct.ProductCode] AS 'ItemP'
```

This query fills the ItemP user field with the item code of the selected product when the Serial/Batch button is pressed. As the ItemP user field is a 'Protected' field, the field value can be used in custom queries for events of the next screen.

Query name: *BXPPSMobilePDC_ProductBatchNumbersAdvScreen_PBWeigh_click*

```
SELECT
'TextBatchQuantity' as Click$,
(
SELECT [@PMX_SCWR].U_PMX_WEIGH * OWGT.WightInMG / OWGT2.WightInMG
FROM [@PMX_SCWR]
JOIN OWGT ON OWGT."UnitDisply" = [@PMX_SCWR].U_UOMPDC
JOIN OITM ON OITM."ItemCode" = $[ItemP]
JOIN OWGT AS OWGT2 ON OWGT2."UnitDisply" = OITM."InvntryUom"
WHERE U_WORKCENTER= $[CurrentPDCBooking.WorkCenterCode]
)
AS TextBatchQuantity,
```

```
'TextBatch' as Click$
```

Query name: *BXPPSMobilePDC_ProductBatchNumbersAdvScreen_PBWeigh_click_after*

```
SELECT 'TextBatch' as Click$
```

4.4. Material

Query name: *BXPPSMobilePDC_MaterialsAdvScreen_MWeigh_click*

```
SELECT
'TextQuantity' as Click$,
(
SELECT [@PMX_SCWR].U_PMX_WEIGH * OWGT.WightInMG / OWGT2.WightInMG
FROM [@PMX_SCWR]
JOIN OWGT ON OWGT."UnitDisply" = [@PMX_SCWR].U_UOMPDC
JOIN OITM ON OITM."ItemCode" = $[SelectedMaterial.ItemCode]
JOIN OWGT AS OWGT2 ON OWGT2."UnitDisply" = OITM."InvntryUom"
WHERE U_WORKCENTER= $[CurrentPDCBooking.WorkCenterCode]
)
AS TextQuantity,
'TextItem' as Click$
```

Query name: *BXPPSMobilePDC_MaterialsAdvScreen_MWeigh_click_after*

```
SELECT 'TextItem' as Click$
```

4.5. Material batch

Query name: *BXPPSMobilePDC_MaterialsAdvScreen_ButtonSerialBatch_click*

```
SELECT $[SelectedMaterial.ItemCode] AS 'ItemM'
```

Query name: *BXPPSMobilePDC_MaterialBatchNumberPickerAdvScreen_MBWeigh_click*

```
SELECT
'TextBatchQuantity' as Click$,
(
SELECT [@PMX_SCWR].U_PMX_WEIGH * OWGT.WightInMG / OWGT2.WightInMG
FROM [@PMX_SCWR]
JOIN OWGT ON OWGT."UnitDisply" = [@PMX_SCWR].U_UOMPDC
JOIN OITM ON OITM."ItemCode" = $[ItemM]
JOIN OWGT AS OWGT2 ON OWGT2."UnitDisply" = OITM."InvntryUom"
WHERE U_WORKCENTER= $[CurrentPDCBooking.WorkCenterCode]
)
AS TextBatchQuantity
```

Query name: *BXPPSMobilePDC_MaterialBatchNumberPickerAdvScreen_MBWeigh_click_after*

```
SELECT 'TextBatch' as Click$
```

5. Customized production data collection process

Process the PDC booking as described in [Production Data Collection](#).

5.1. Stop Job/Partial Job screen

After pressing the Stop or the Partial button, the user is prompted to the Stop Job/[Partial Job screen](#). An additional Weigh button is displayed on the screen. Press this button to weigh the produced quantity instead of adding it manually for the main product.

If the main product is managed by batches and multiple batches were produced in this job, weigh every batch on this screen.

The screenshot shows the 'Partial Job' screen in the Mobile PDC application. The header includes the application name 'Mobile PDC', the user 'TEST_WMSMF (PMX_BUDTOSH2) - John Doe', and the date '10/12/17 03:06 PM'. The main content area displays the following information:

Production Order	#136 pMP1101 (Main product)	UoM	g
Operation	136-1 (OPP1 - Operation)		
Started	10/11/17 01:38 PM	<input type="checkbox"/> Completed	
Bin Location			
Duration	1,526 min	This Day	
Quantity	200	Rejected Quantity	0

At the bottom of the screen, there are three buttons: 'Done' (F1), 'Cancel' (ESC), and 'Weigh'. The 'Weigh' button is highlighted with a red box.

5.2. Products screen

On the [Products screen](#) the main product and by-products can be received.

To add the produced quantity for the by-product by weighing, select the by-product and press the Weigh button. If the by-product is managed by batches and multiple batches were produced in this job, weigh every batch on this screen.

Please note: It is not possible to modify the produced quantity for the main product on this screen.

Mobile PDC TEST_WMSMF (PMX_BUDTOSH2) - John Doe 10/11/17 12:40 PM

[Products]

Production Order #136 pMP1101 (Main product) UoM g

Operation 136-1 (OPP1 - Operation)

Product SSCC

Quantity 412

Item	Name	SSCC	Quantity
pMP1101	Main product		200 of 200 g
pBP011	By-product (batch)		412 of 400 g

Done Cancel Serial / Batch Weigh

Scale weigh result

#	Code	Name	Weigh	Uom	Work center	Uom for PDC
1	scale01		425		wWC	g
2	scale02		0.412		wAS	kg
3	scale03		0.379		wCT	kg
4						

OK Cancel

5.3. Product, by-product batch

If the (by)-product is managed by batches, select the (by)-product and press the Serial/Batch button in order to add the batch numbers. On the [Product Batch screen](#) scan the batch number then press the Weigh button to add the quantity by weighing.

Please note: The combined quantity of the batches must be equal to the (by)-product quantity specified on the previous screen.

Mobile PDC TEST_WMSMF (PMX_BUDTOSH2) - John Doe 10/12/17 03:24 PM

Product Batch Numbers

Production Order #136 pMP1101 (Main product) UoM g

Operation 136-1 (OPP1 - Operation)

Item pBP011 (By-product (batch)) SSCC

Batch Number Batch Number

Quantity 204 [Best Before]

Batch Number	SSCC	Quantity	[Total Quantity]	Batch Number	[Best Before]
BP171010		208	208		1/1/0001 12:00:00 AM
BP171011		204	204		1/1/0001 12:00:00 AM

Quantity 412 g Of 412 g
Rejected Quantity 0 g Of 0 g

Done Cancel Rejected **Weigh** Delete

5.4. Material

On the [Materials screen](#) materials to be consumed can be reported.

Select the material and press the Weigh button in order to add the quantity by weighing. If the material is managed by batches and multiple batches are going to be consumed in this job, weigh every batch on this screen.

Mobile PDC TEST_WMSMF (PMX_BUDTOSH2) - John Doe 10/12/17 03:34 PM

Materials

Production Order #136 pMP1101 (Main product) UoM g

Operation 136-1 (OPP1 - Operation)

Item Bin Location

Quantity 2,230

Item	Name	Bin Location	Quantity
mM3475	Material		2,230 of 3,000 g

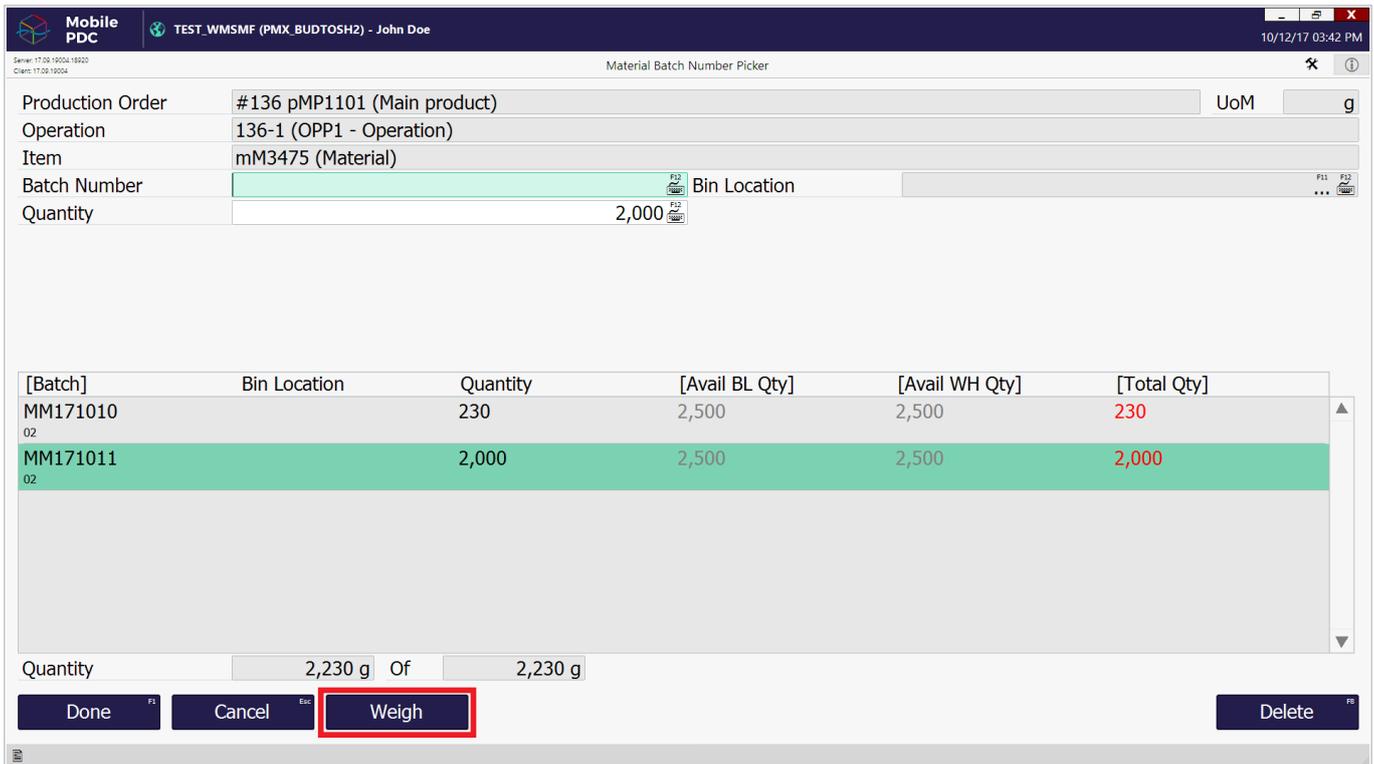
Warehouse: 02

Done Cancel Serial / Batch **Weigh** Delete

5.5. Material batch

If the material is managed by batches, select the material and press the Serial/Batch button in order to add the batch numbers. On the **Material Batch screen** scan the batch number then press the Weigh button in order to add the quantity by weighing.

Please note: The combined quantity of the batches must be equal to the material quantity specified on the previous screen.



Translation Customization

Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

The translation customization feature can be used for creating custom translations. It is also possible to create custom translation for a given SBO user only.

Open the Translation customization via the following path: Tools > Produmex Manufacturing > Developer Tools > Language BxDictionary > Translation Customization



1. Change the value on the Translation Customization form

1.1. Find the value

Add the text you would like to customize as the 'Value'. Only the exact search text will be matched. To search based on partial text, use asterisk (*) as a wildcard.

Space characters will be considered. The filter is not case sensitive on MSSQL.

Example:

- Search for 'allocation window'

The screenshot shows the 'Translation Customization for Produmex Manufacturing' interface. The search criteria are: Key (empty), Value (allocation window), Language (English), Customized (Both), User Code (manager), and Resource File Name (-). The 'Display Original' checkbox is checked. The results table is as follows:

Key	Value	Comments	User Code	Language	Customized	Resource File Name
AllocationWindow	Allocation Window		*	en-US	<input type="checkbox"/>	L
BXPANUFOPER_BXPAINd	Allocation Window		*	en-US	<input type="checkbox"/>	D
BXPPRODORDEROPER_BXPAINd	Allocation Window		*	en-US	<input type="checkbox"/>	D
BXPPRODTREEOPER_BXPAINd	Allocation Window		*	en-US	<input type="checkbox"/>	D

- Search for 'window'

The screenshot shows the 'Translation Customization for Produmex Manufacturing' interface. The search criteria are: Key (empty), Value (window), Language (English), Customized (Both), User Code (manager), and Resource File Name (-). The 'Display Original' checkbox is checked. The results table is empty.

Key	Value	Comments	User Code	Language	Customized	Resource File Name
-----	-------	----------	-----------	----------	------------	--------------------

- Search for 'allocation win*'

The screenshot shows the 'Translation Customization for Produmex Manufacturing' interface. The search criteria are: Key (empty), Value (allocation win*), Language (English), Customized (Both), User Code (manager), and Resource File Name (-). The 'Display Original' checkbox is checked. The results table is as follows:

Key	Value	Comments	User Code	Language	Customized	Resource File Name
AllocationWindow	Allocation Window		*	en-US	<input type="checkbox"/>	L
AllocationWindowDeflength	Allocation window def.length		*	en-US	<input type="checkbox"/>	L
AllocationWindowTimeCannotBeNegative	Allocation Window Time cannot be negative		*	en-US	<input type="checkbox"/>	L
AllocationWindowTimeIsNegativeForOperation	Allocation Window Time is negative for operation {0}		*	en-US	<input type="checkbox"/>	L
AllocationWindowUOM	Allocation window UOM		*	en-US	<input type="checkbox"/>	L
AllocationWindowUoMHasNotBeenDefinedForOperation	Allocation Window UoM has not been defined for operation {0}		*	en-US	<input type="checkbox"/>	L
AllocationWindowUoMNotDefined	Allocation Window UoM has not been defined		*	en-US	<input type="checkbox"/>	L
BXPANUFOPER_BXPAINd	Allocation Window		*	en-US	<input type="checkbox"/>	D
BXPANUFOPER_BXPWnUoM	Allocation Window UoM		*	en-US	<input type="checkbox"/>	D
BXPPRODORDEROPER_BXPAINd	Allocation Window		*	en-US	<input type="checkbox"/>	D
BXPPRODORDEROPER_BXPWnUoM	Allocation Window UoM		*	en-US	<input type="checkbox"/>	D
BXPPRODTREEOPER_BXPAINd	Allocation Window		*	en-US	<input type="checkbox"/>	D
BXPPRODTREEOPER_BXPWnUoM	Allocation Window UoM		*	en-US	<input type="checkbox"/>	D

- Search for '* window'



Select the language from the dropdown list. Supported languages:

- cs-CZ: Czech
- de-DE: German
- en-US: English
- es-MX: Spanish (MX)
- fr-FR: French
- hr-HR: Croatian
- hu-HU: Hungarian
- it-IT: Italian
- ja-JP: Japanese

- nl-NL: Dutch
- pt-BR: Portuguese (BR)
- pt-PT: Portuguese
- ru-RU: Russian

The list also can be filtered based on the following:

- Key
- User Code
- Customized (possible values: Yes, No, Both)
- Resource File Name

If you search for a value in a language other than English, and you would like to display the English value as well, check the 'Display Original' checkbox.

Press the 'Find' button to start the search.

Please note: The maximum number of rows to display is 500. If the specified filters add more than 500 lines as a result, only the first 500 lines will be displayed. It is recommended to select more restrictive filters.

To clear the specified filters, click on the 'Clear Filters' button.

Grid

Key

The localization key that will be used for translation.

Value

The translation value. The text that will be displayed on the interface.

User Code

The SBO user code. If a translation is applied for every user, the asterisk (*) is displayed.

Language

The language of the translation.

Customized

Marks whether the translation value is customized or not. If the checkbox is checked, the translation value is customized.

Resource File Name

Identifies the origin of the translation value.

- D & T stand for the translation of database values (eg. user tables, user fields, valid values)
- L & F stand for the translation of localization values (eg. buttons, functions, system messages)
- R stands for the translation of report values

To check the translation key and the resource file name code on the interface, enable 'Change Debug Mode' from Tools > Produmex Manufacturing > Developer Tools >

Language BxDictionary.

Translation values are displayed in the following format: (*resource file name:key: value*)



Please note: the key and resource file name code is not displayed on standard SBO documents.

1.2. Add the custom value

Click in the value field and enter the custom value. You can specify the user code and add a comment as well.

Click on the 'Update' button.

In order to apply the changes, restart the Produmex Manufacturing add-on.

1.3. Remove values

Right-click on the row that you would like to remove and select the 'Delete row' option. Click on the 'Update' button to apply the changes.

2. Import custom values

Custom translation values can be imported from a .csv file. Click on 'Import Translation Customization from CSV' option from Tools > Produmex Manufacturing > Developer Tools > Language BxDictionary. Then browse the .csv file.

After a successful import, the system reloads the translation files and displays the following message:

"Translation files reloaded. Current language is en-US. "

File structure

- Resource File Name: Identifies the origin of the translation value.
- Key: The localization key.
- Language: The language code.
- Value: Add the custom translation value here.
- Comments: Remarks can be added here. It is possible to leave this field empty.
- Customized: Marks whether the translation value is customized or not. Possible values: True/False.
- User code: The SBO user code. To add a translation for every user, add *.

- Add-On: The identifier of the add-on. Add 'BXPPS'.

Use TAB as the separator between the columns in the .csv file.

Please note: Both the resource file name and the key are needed in order to identify the object. We recommend to check the resource file name and key on the Translation Customization for Produmex Manufacturing form or to export the Translation Customization table to a .csv file and customize the values in the exported .csv file.

Example .csv file:

Resource File Name	Key	Language	Value	Comments	Customized
User Code	Add On				
D BXPABSRSN	en-US	AbsReason	True	* BXPPS	
L BoMTree	en-US	Product structure	For manager only		True
manager	BXPPS				

Translation Customization for Produmex Manufacturing

Key: Language: Find

Value: Customized: Clear Filters

User Code: Resource File Name:

Display Original:

Key	Value	Comments	User Code	Language	Customized	Resource File Name
BoMTree	BoM Tree		*	en-US	<input type="checkbox"/>	L
BoMTree	Product structure	For manager only	manager	en-US	<input checked="" type="checkbox"/>	L



3. Export Translation Customization table

To export the Translation Customization table to a .csv file, select 'Export Translation Customization table to CSV' from Tools > Produmex Manufacturing > Developer Tools > Language BxDictionay.

The entire table will be exported. It is not possible to only export selected lines of the table.

From: <https://wiki.produmex.name/> - **Produmex**

Permanent link: <https://wiki.produmex.name/doku.php?id=implementation:manufacturing:customizationcomplete&rev=1552995684>

Last update: **2019/03/19 11:41**

