

Produmex Manufacturing Customization Guide

Customization Technology

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

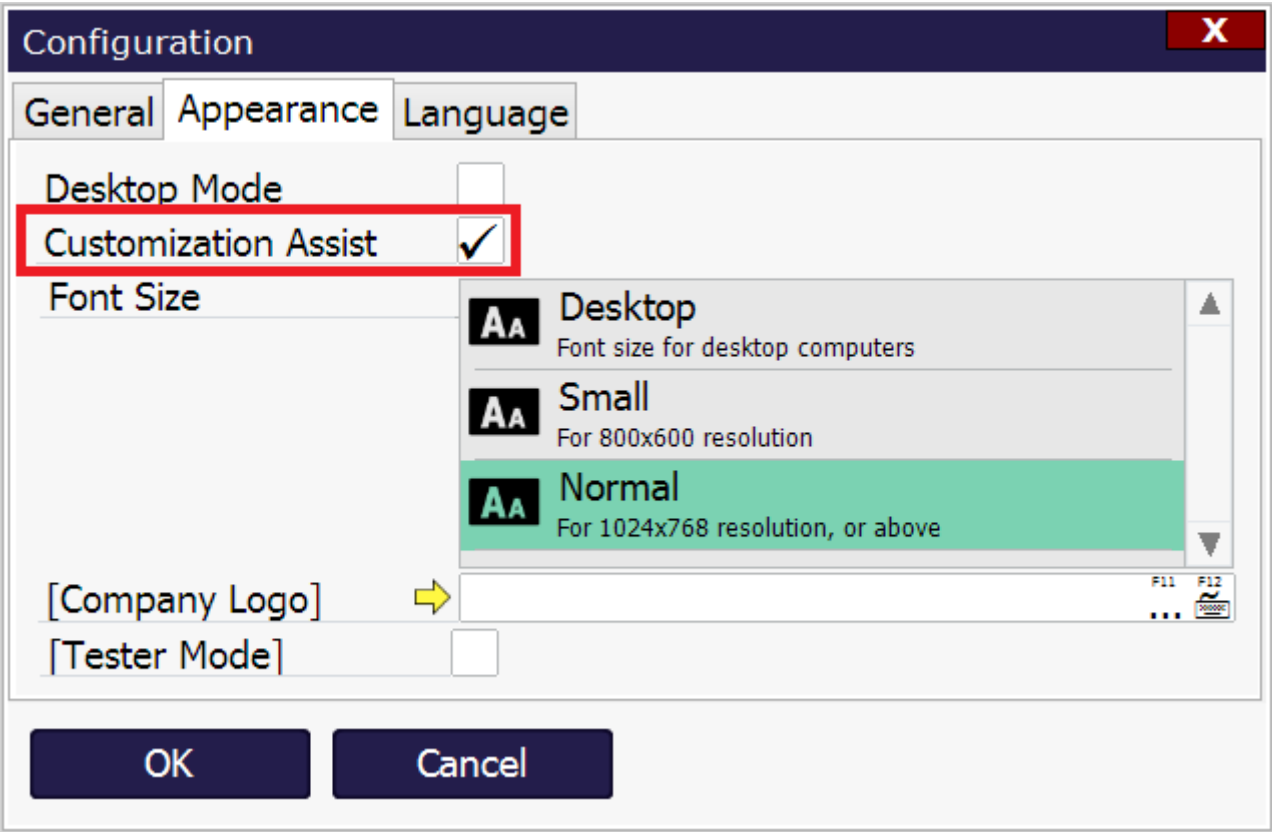
1. Configure and run client application

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.

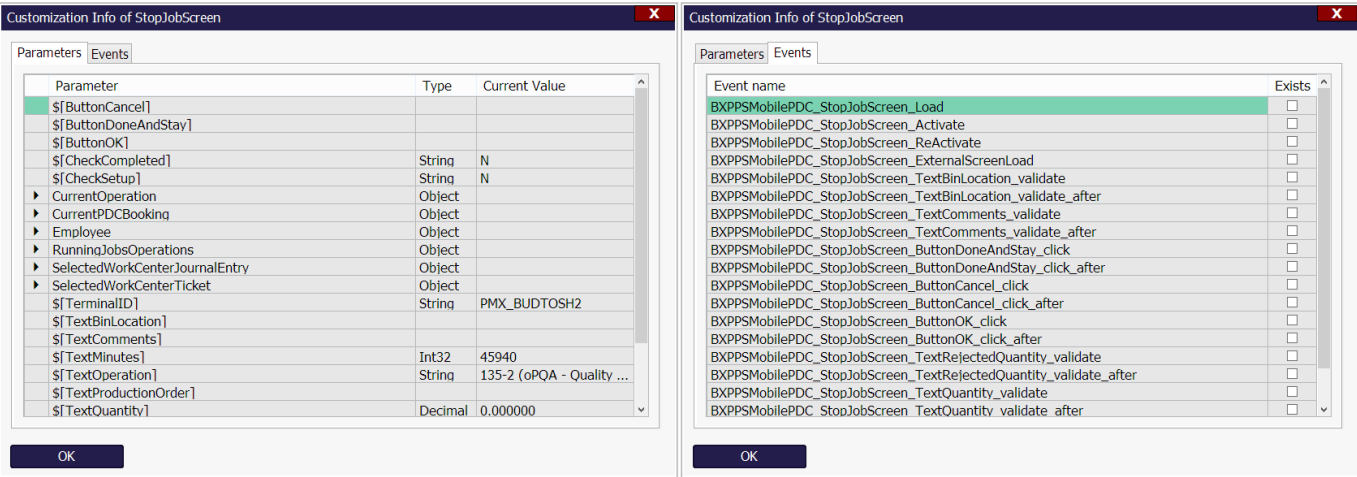
Start the Produmex PDC Configuration application. On the '[Appearance](#)' tab tick the Customization Assist checkbox to enable the Customization Assist mode.



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 Scan Customization Examples 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.	Produmex Scan: <i>BXMobileWH9</i> Produmex PDC: <i>BXPPSMobilePDC</i>
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
Position Data	Object placement can be changed in this field.	x: - horizontal position (% of screen width from left) y: - vertical position (% of screen height from top) w: - width (% of screen width) h: - height (% of screen height from top) f: - font size (pixel) only on 'Button' Only integer values are allowed. Separate the different parameters with the ';' character. Example: x:1;y:70;w:50;h:9
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 'BXMobilWH9_GoodsReceiptPOScreen_Load'

For information about the input values and results please see the related [Customization example](#).

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**

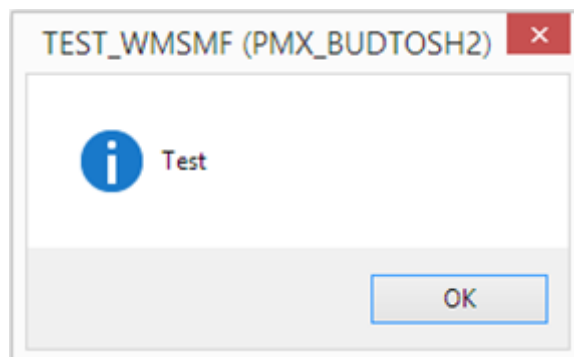
A blue line will appear at the footer of the screen with the message. The event will execute.

- **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.

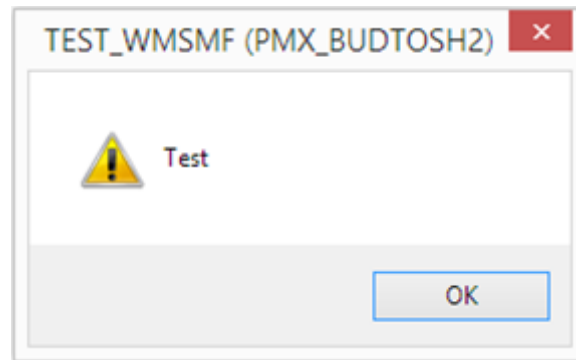
- **IM: Information with message box**

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



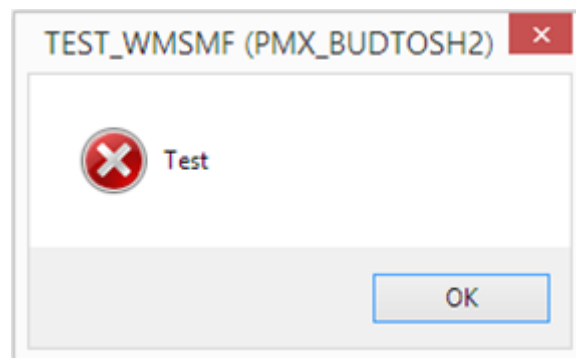
- **WM: Warning with message box**

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



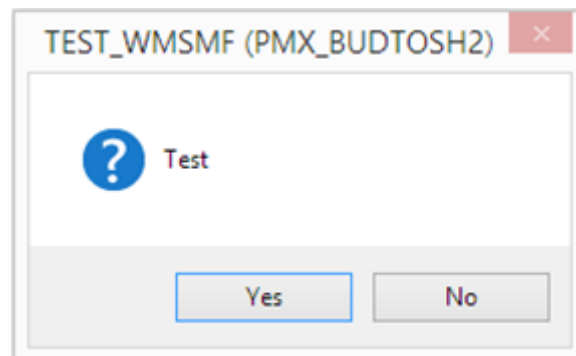
- **EM: Error with message box**

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

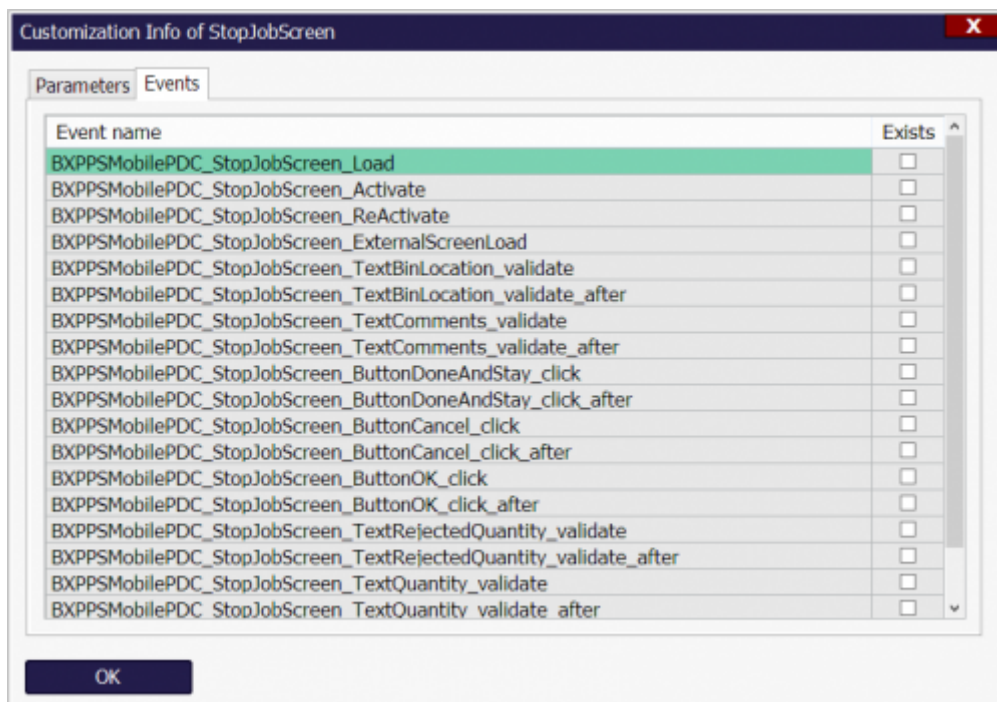
1. Preset field data

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.

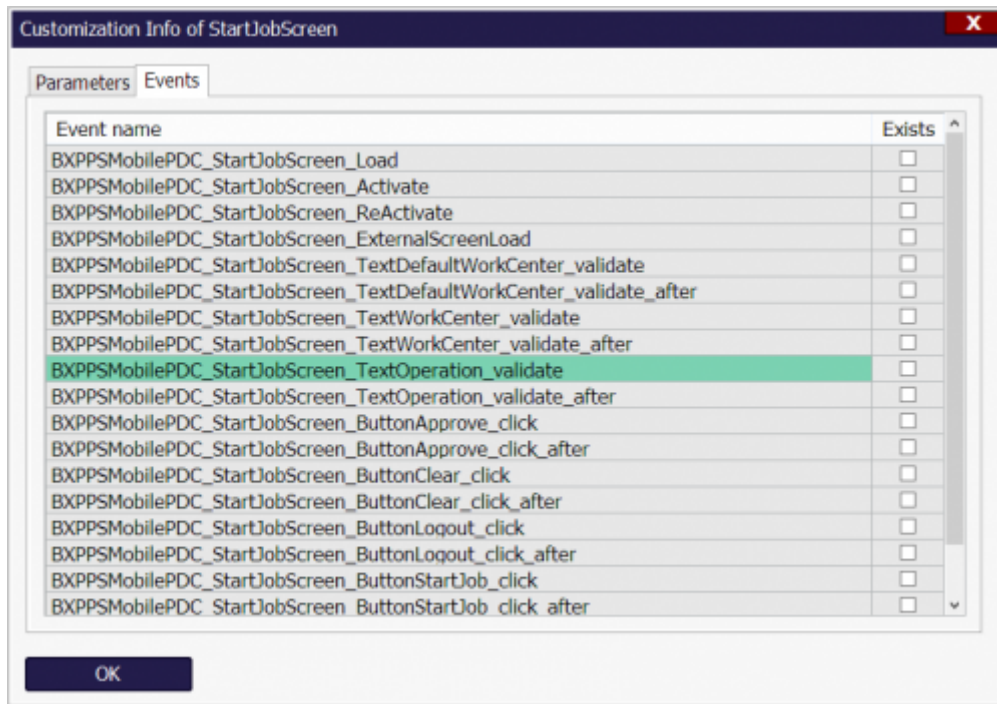
Please note: *You only need to restart the application once a new query has been added. If you only add modifications to the user query, Produmex PDC doesn't have to be restarted.*

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

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:

Mobile PDC TEST_WMSMF (PMX_BUDTOSH2) - John Doe 06/20/17 04:13 PM

Server: 17.05.31007.18920 Client: 17.05.31007

Start Job

Operation 56-1 (oPAS - Bike Assembly - 56)

Work Center wAS (Assembler Team)

Start Setup F1 Start Job F2 Admin F3 Clear F4 Logout Esc

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*

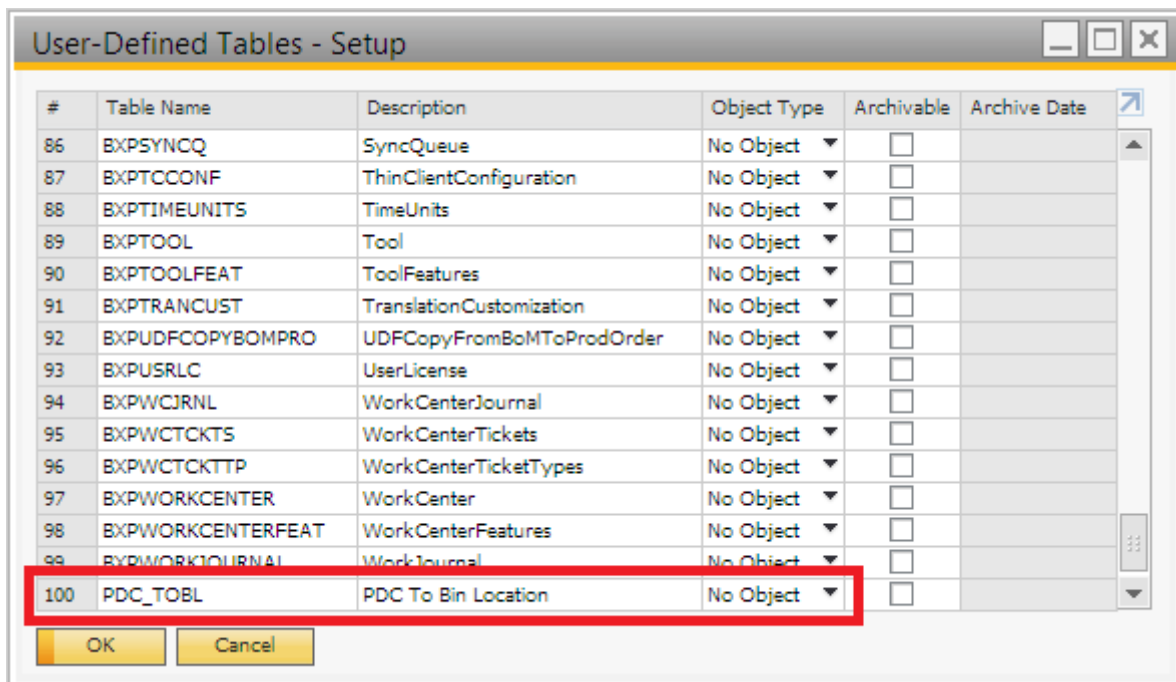
```
IF $[TextEmployeeID] = ''
SELECT 'Scan Employee ID' AS 'Message$', 'E' AS 'MessageType$'
```

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

6. Save the Destination Bin Location for the employee

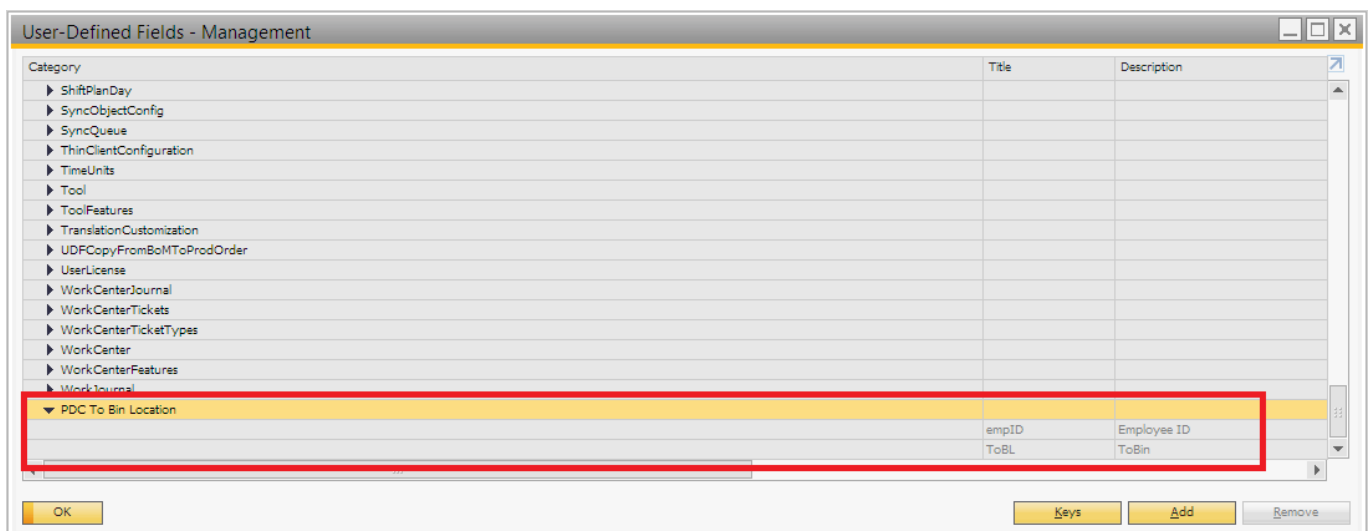
This customization will save the last added bin location on the Products screen. When this screen next loads to the employee, the Bin Location code will be automatically filled. This way if the employee uses the same destination bin location, (s)he does not have to re-enter it.

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

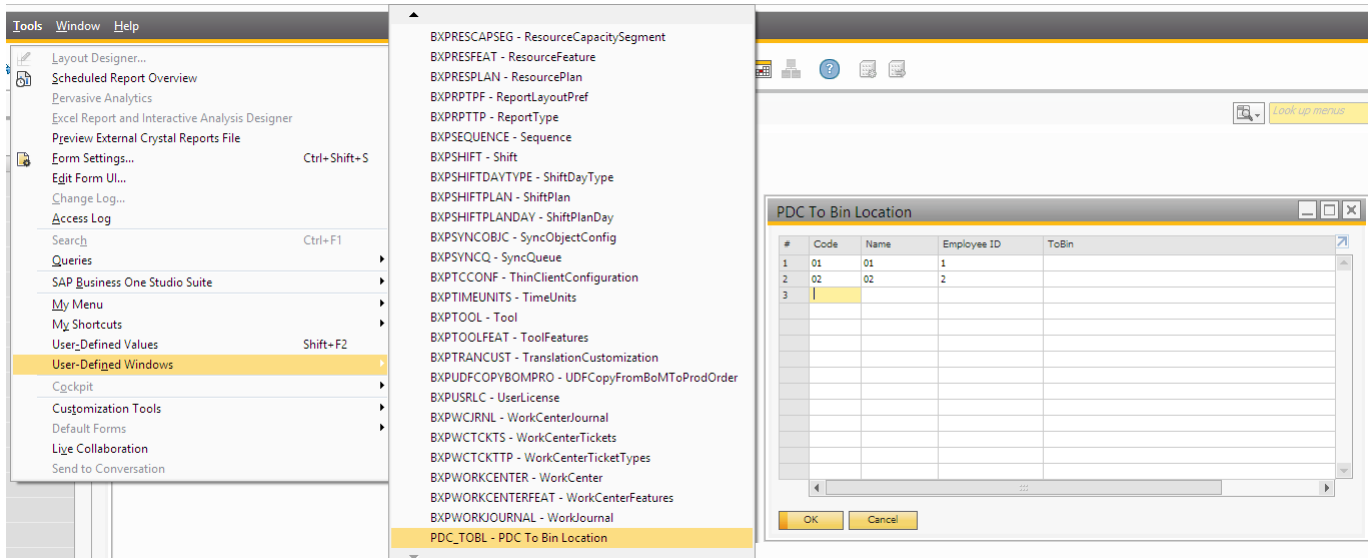


Add the following fields to the user table: [Tools>Customization Tools>User Defined Fields-Management]

- 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 offers an easy solution for report customization with Crystal Reports.

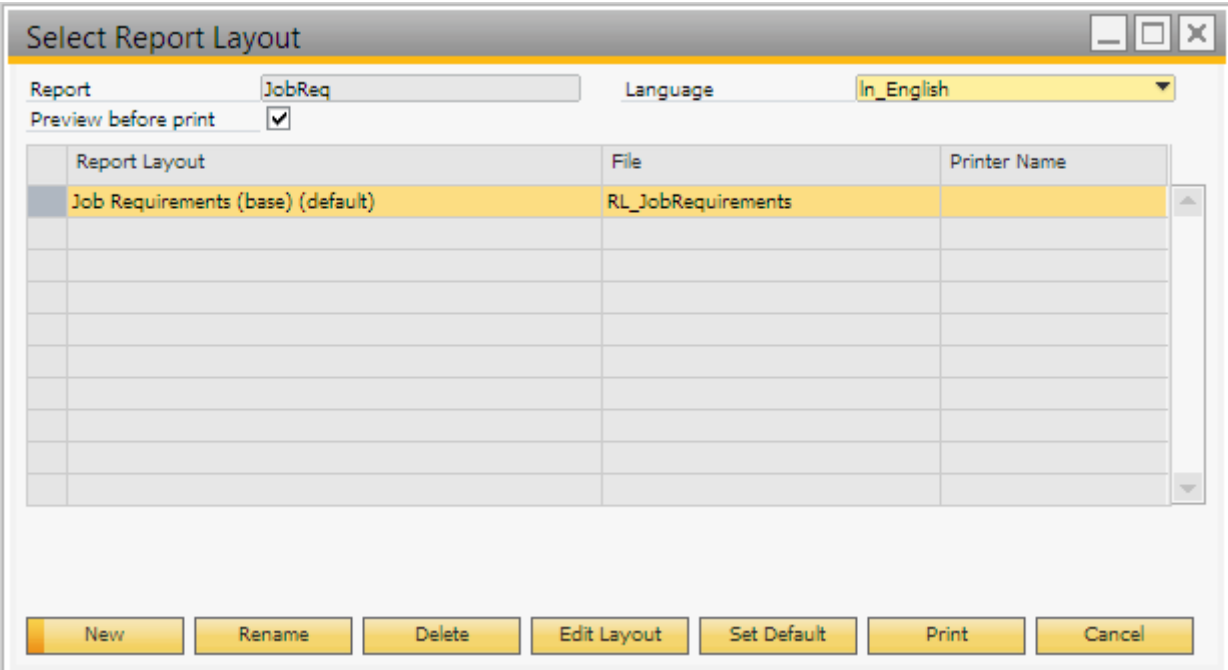
Please note: This document describes the custom report creation procedure in MSSQL.

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

Operation:	oPAS - Bike Assembly	Operation ID: 00048789
	Begin Date&Time: 02/19/17 04:12 PM	Production Order: 579 / 4
	Before Time: 0.00 [min]	Product Code: p1001-1 - Red Bike
	After Time: 0.00 [min]	
	mM1001 Painted Bike Framework	1.00
	00048786 	

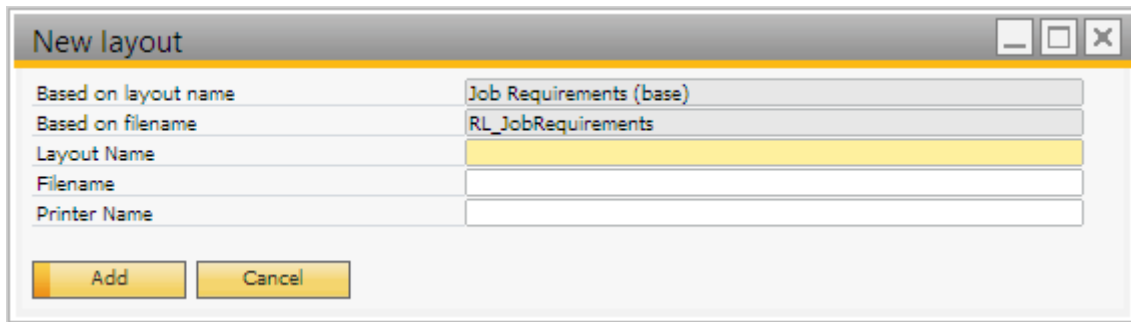
Open the report. On the 'Select Report Layout' form select a base layout on the grid then click on the 'New' button to create a new layout.

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.

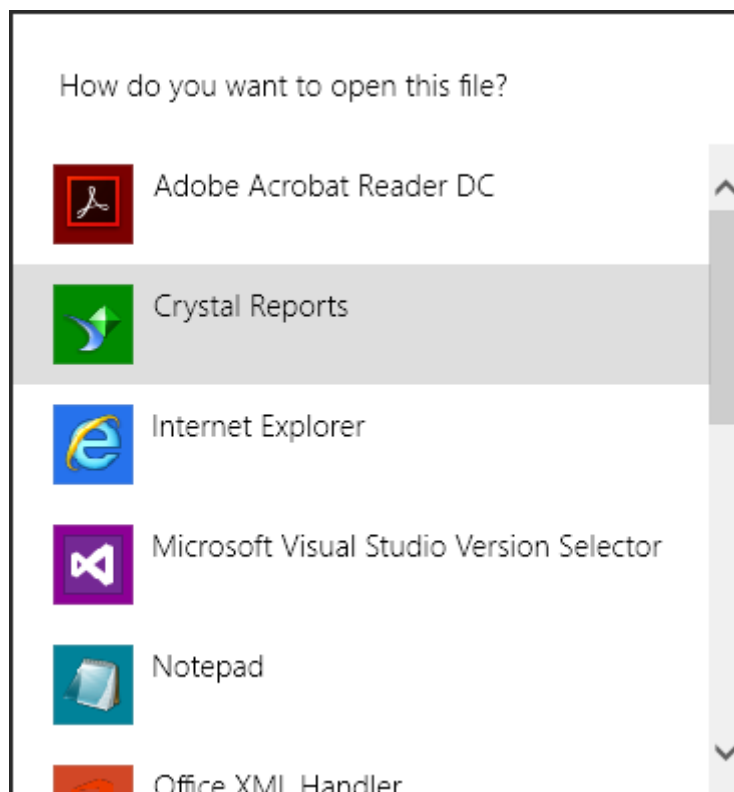


A dialog box titled "New layout" with a standard Windows window border. It contains four input fields: "Based on layout name" with the value "Job Requirements (base)", "Based on filename" with the value "RL_JobRequirements", "Layout Name" (empty), and "Filename" (empty). Below these is a "Printer Name" field. At the bottom are two buttons: "Add" and "Cancel".

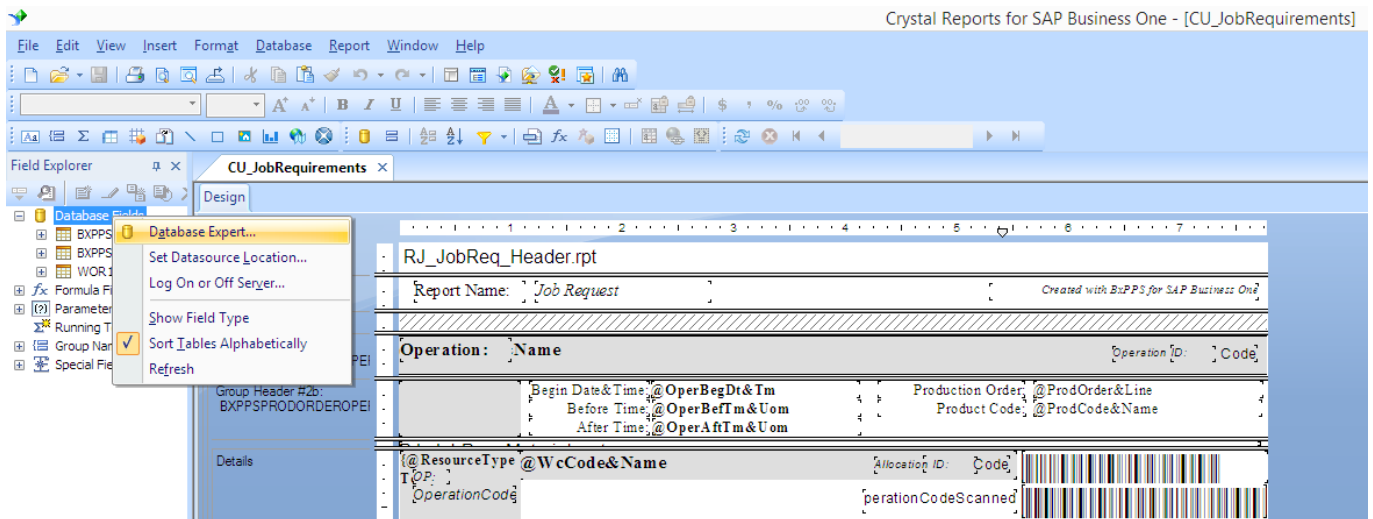
Based on layout name	Job Requirements (base)
Based on filename	RL_JobRequirements
Layout Name	
Filename	
Printer Name	

Add Cancel

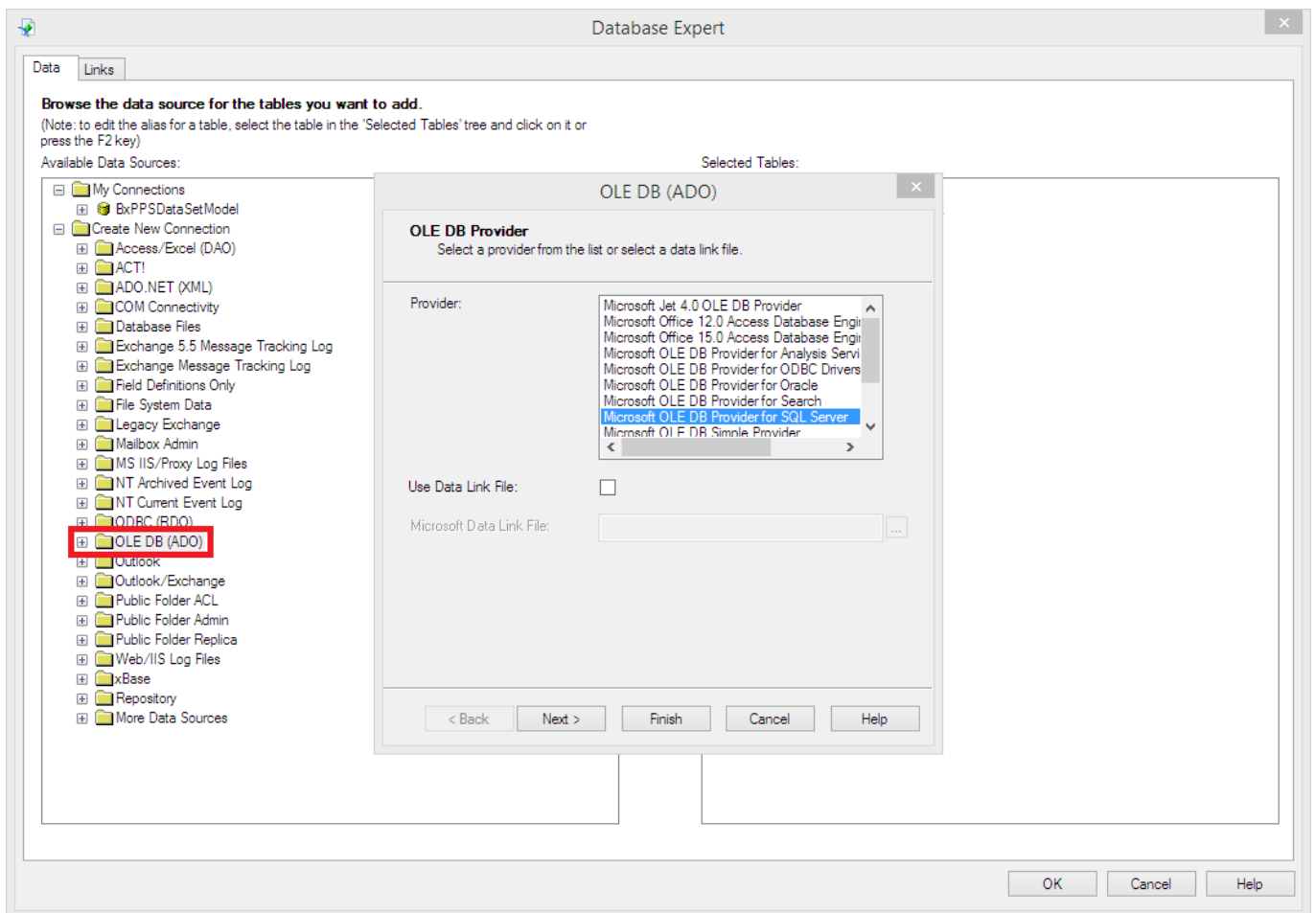
After adding the custom layout, select it on the grid. Click on the 'Edit Layout' button and open the layout with Crystal Reports.



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.



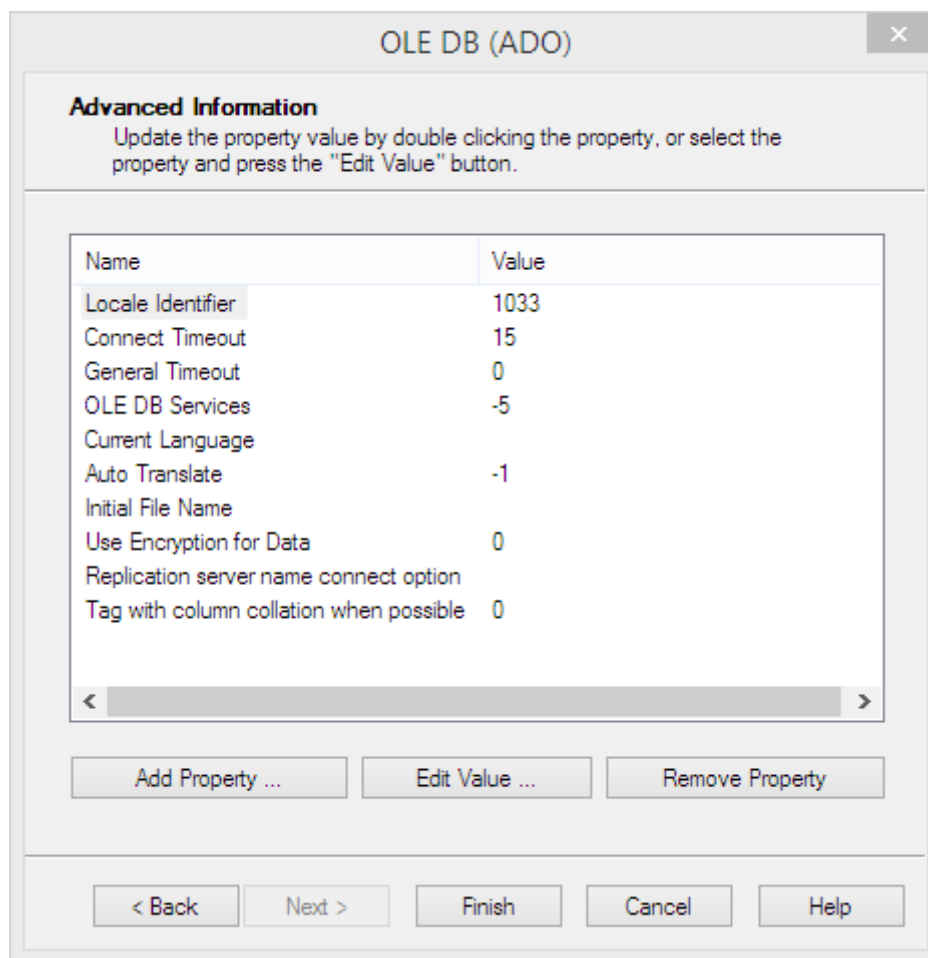
On the next screen add the connection information. Select the local server as the 'Server' and enter the User ID and Password. Then select the database from the dropdown menu and click on 'Next'.

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.

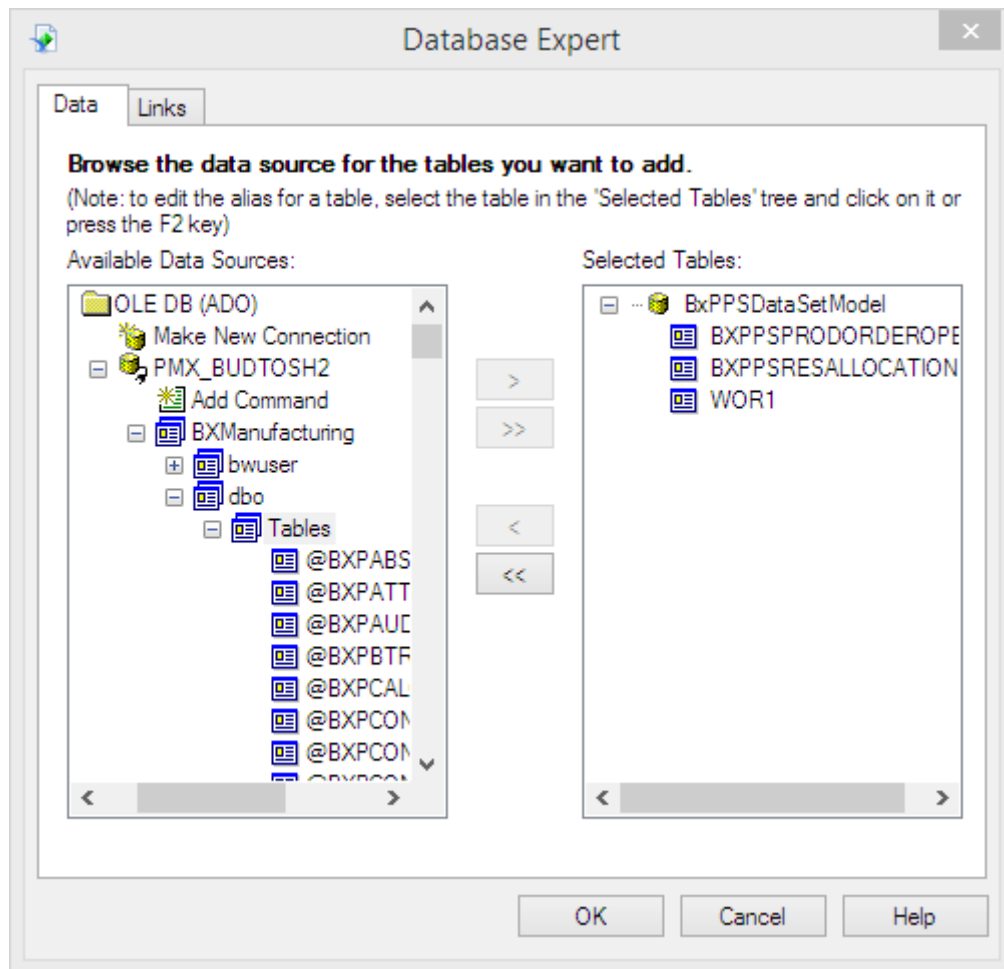


The image shows a Windows-style dialog box titled "OLE DB (ADO)". Inside the dialog, there is a section titled "Connection Information" with the instruction "Provide necessary information to log on to the chosen data source." Below this, there are five fields: "Server:" with a dropdown menu showing "PMX_BUDTOSH2"; "User ID:" with a text box containing "sa"; "Password:" with a text box containing eight dots; "Database:" with a dropdown menu showing "Manufacturing"; and "Integrated Security:" with an unchecked checkbox. At the bottom of the dialog, there are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help".

On the next screen click on 'Finish'.

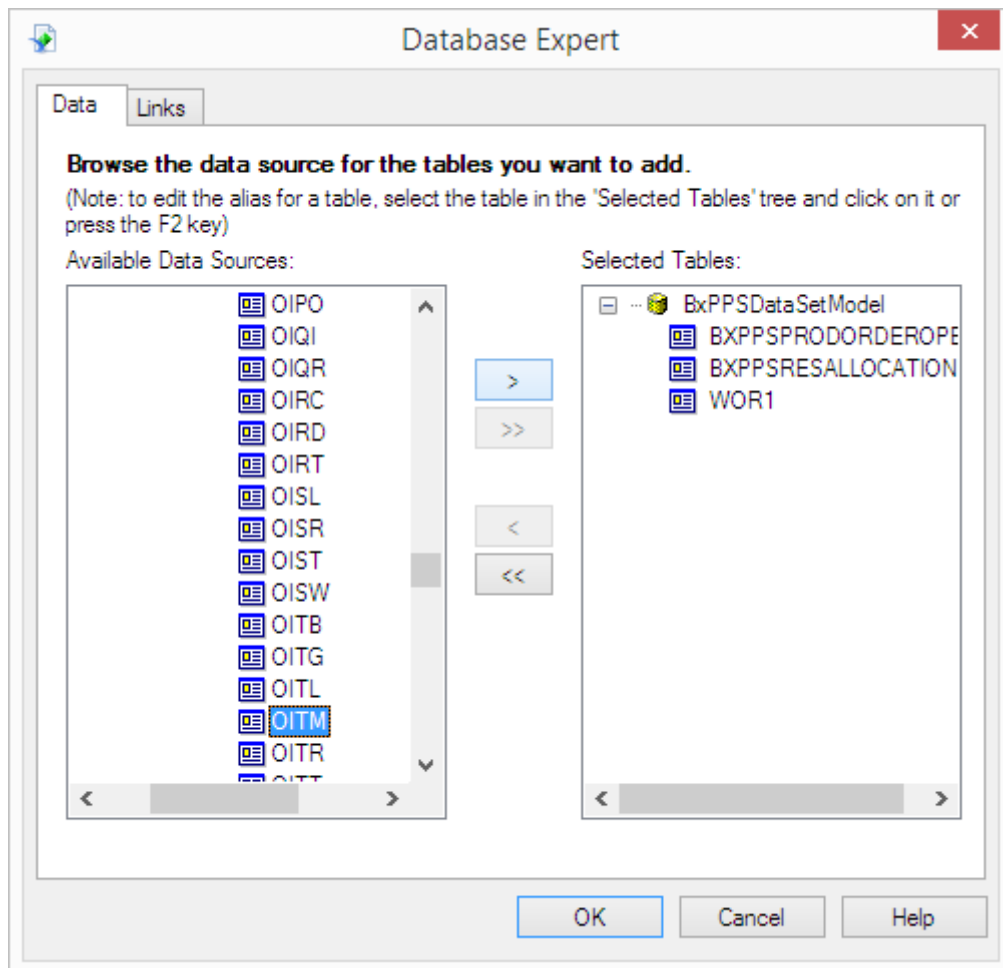


The database is added to the OLE DB folder.



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\BX Manufacturing\BxPPSDataSetModel.xsd*

ADO.NET (XML)

Connection
Please enter connection information...

File Path : PPS\BxPPSDataSets\BxPPSDataSetModel.xsd ...

Class Name: [Dropdown]

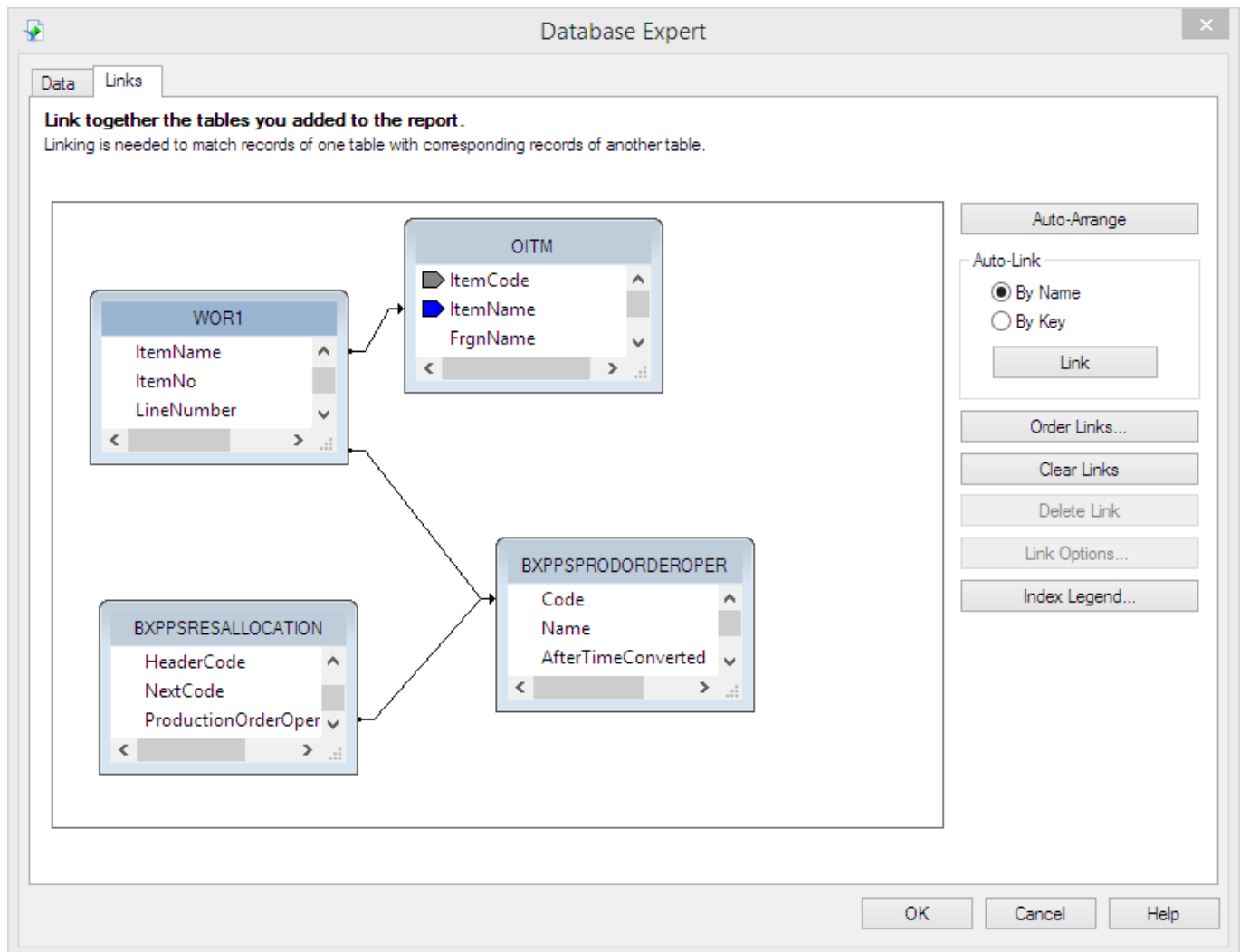
Use DataSet from Class: ☐

< Back Next > Finish Cancel Help

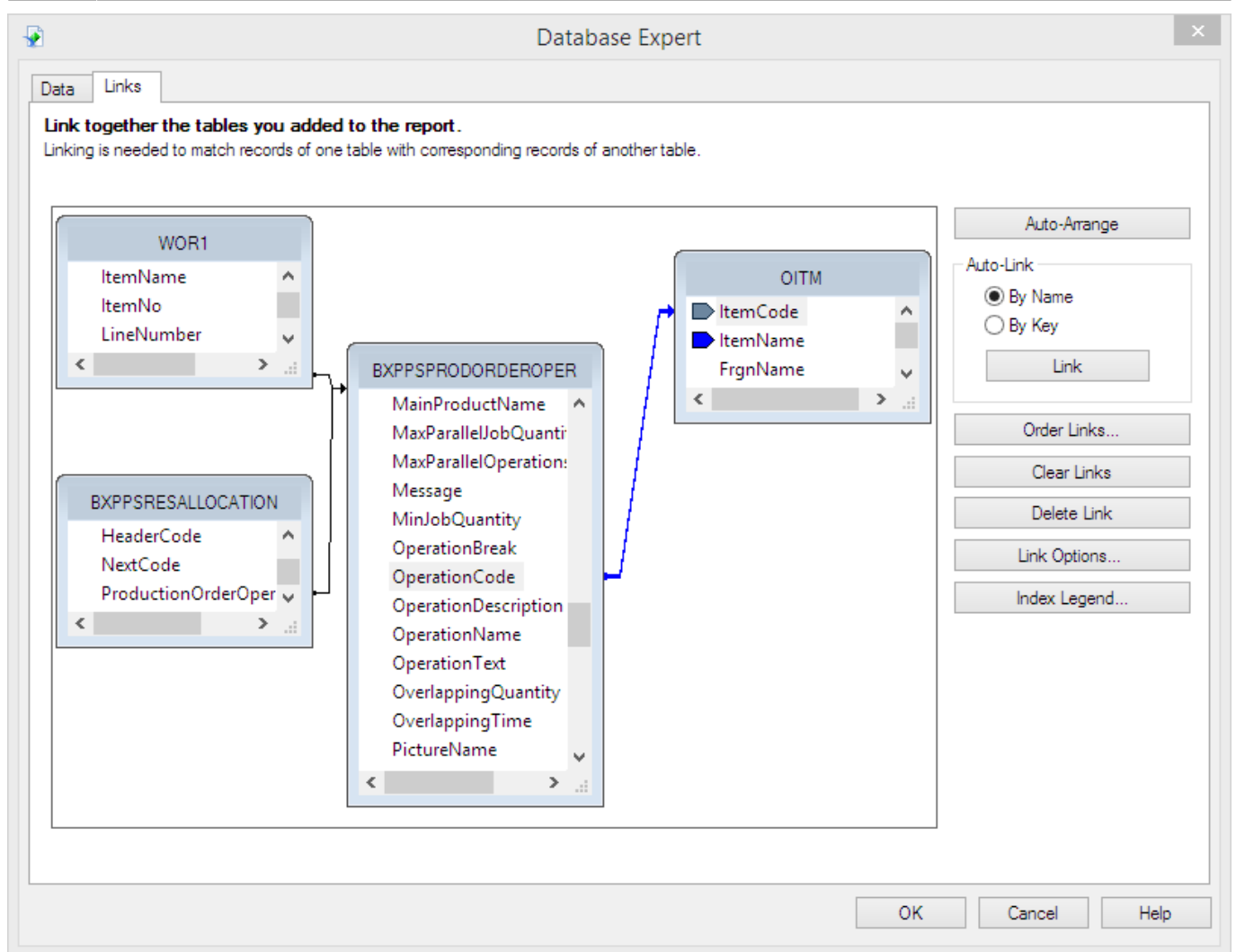
Data Links

Browse the data
(Note: to edit the a press the F2 key)
Available Data Sources

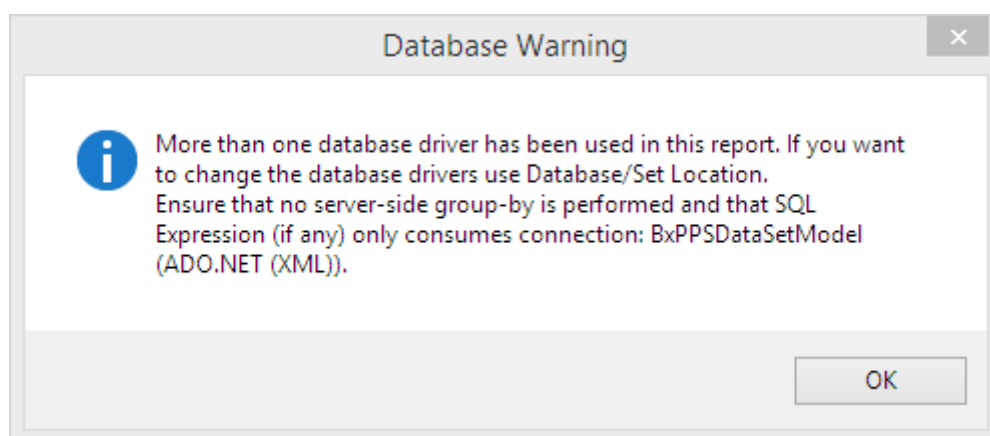
On the Links tab the links between the table records are shown. These links are created automatically and might not be correct for the customization goals. Correct the links by deleting the wrong ones and creating new links. Click on the 'OK' button.



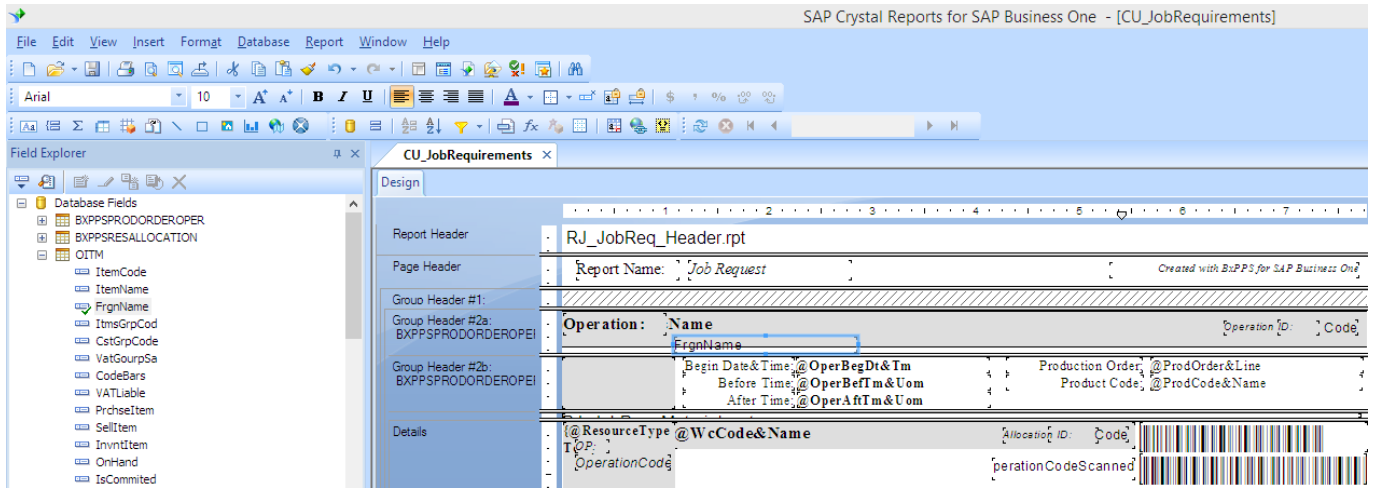
In the example the 'Item Name' field from the OITM table was linked automatically to the WOR1 table. First we delete this link then we create the new one by connecting the Operation Code from the BXPPSPRODORDEROPER table to the Item Code on the OITM table.



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.

Operation:	oPAS - Bike Assembly	Operation ID: 00048789
	Assembly du bike	
	Begin Date&Time: 02/19/17 04:12 PM	Production Order: 579 / 4
	Before Time: 0.00 [min]	Product Code: p1001-1 - Red Bike
	After Time: 0.00 [min]	
	mM1001 Painted Bike Framework	1.00
	00048786	

Special customization

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.emplID)
- [%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.emplID)
- [%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.emplID)
- [%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.

Set up Password Protected Login for Produmex PDC

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.

Last update: 2017/09/29 11:27 implementation:manufacturing:customizationcomplete <http://wiki.produmex.name/doku.php?id=implementation:manufacturing:customizationcomplete>

Produmex Manufacturing Settings

General SQL Logs Reports MRP PDC Prod.Order Master Data MTO Thin Client Thin Client 2 Food Scheduled Realocator Custom Settings

Is Serial Batch PDC Enabled ☒
800x600 PDC Wizard ☐
HD resolution PDC Wizard ☐
PDC Use Direct State ☒
PDC Allow Undo ☐
PDC Undo Only No Transaction ☐
PDCTooGoodPerformanceThreshold 1.000000
PDCTooBadPerformanceThreshold 1.000000
PDC book duration in Break bookings ☐
Use Operation Parameters ☐
GoodIssue Series Name
ReceiptFromProd. Series Name
IssueForProduction Series Name
GoodReceipt Series Name
PDCProcessor Last Run Date 07/28/17
PDCProcessor Last Run Time 1422
PDCProcessor Retry Times 5:15:60
Managing Rejected Batched PDC Transactions ☐
Default Production Warehouse for Managing Rejected Batched PDC Transactions
Default Base Item Warehouse for Managing Rejected Batched PDC Transactions
Employee Card Code Field U_PDCPass
Default Attendance Model
PDCProcessor Block Start Date
PDCProcessor Block Start Time 0
PDCProcessor Block Minutes 0
Profit Center for Managing Rejected Batched PDC Transactions
Account Code for Managing Rejected Batched PDC Transactions
Use Production Order Doc Entry and Line For PDC ☐
No Work Center Change with Use Production Order Doc Entry and Line For PDC ☐
PDC Delete Allocation Last Date
FlexProdScheduler Last Run Date
FlexProdScheduler Last Run Time 0
Enable Lost Allocations in PDC Shop/Floor Wizard ☐
PDC Processor Interval Time 30
PDC Material Batch Overbooking Permission WVOverbookingPermission_NotAllowed

OK Cancel

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

Produmex Manufacturing Settings

General SQL Logs Reports MRP PDC Prod.Order Master Data MTO Thin Client Thin Client 2 Food Scheduled Realocator Custom Settings

Worker can modify bookings ☐
Approver can modify bookings ☐
Global idle timeout (seconds) 0
Global screen timeout (seconds) 0
Employee approver role Approver
Employee Workshop Monitor Role Workshop Monitor
Employee Quality Control Role QC Inspector
Workcenter Admin Role
Enable PDC ☒
Enable PTM ☒
Enable QC ☒
Enable Workshop Monitor ☒
Enable Workcenter Journal ☒
Enable Workcenter Tickets ☒
Enable Legacy Mode in PDC ☐
Pre-fill planned material quantities ☒
Pre-fill planned by-product quantities ☒
Pre-fill the bin locations quantities with available quantities ☐
Skip material quantities screen ☐
Skip by-product quantities screen ☐
Skip material serial/batch quantities screen ☐
Skip product serial/batch quantities screen ☐
Logout after PDC bookings ☐
Enable Partial Book & Stay ☐
Can insert new materials into production orders ☐
Login Is Password Protected ☒
Only Job Bookings On Running Jobs Screen ☐
Force enter product serial/batch numbers and quantities ☐

Update Cancel

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

Mobile PDC TEST_WMSMF (PMX_BUDTOSH2) 08/07/17 03:03 PM

Server: 17.05.31007.18920 Client: 17.05.31007

Login

Employee John Doe

From:

<http://wiki.produmex.name/> -

Permanent link:

<http://wiki.produmex.name/doku.php?id=implementation:manufacturing:customizationcomplete> 

Last update: **2017/09/29 11:27**