Customization Examples

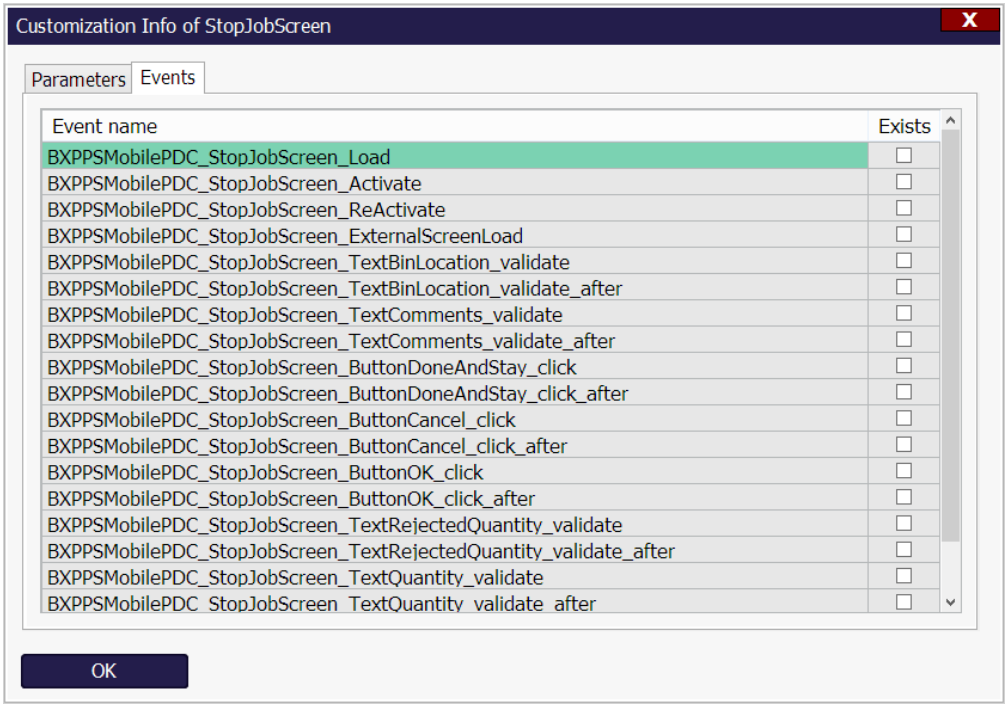
Produmex Manufacturing is a legacy product and Boyum IT Solutions no longer sells new installations for it.

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_BXPPrOOI =
$[CurrentOperation.Code] AND U_BXPIsUnd = 'N') >
$[CurrentOperation.PlannedQuantity]
THEN 0
ELSE $[CurrentOperation.PlannedQuantity] - ISNULL( (SELECT SUM(U_BXPCoQty)
FROM [@BXPPDCBOOKING] WHERE U_BXPPrOOI = $[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.

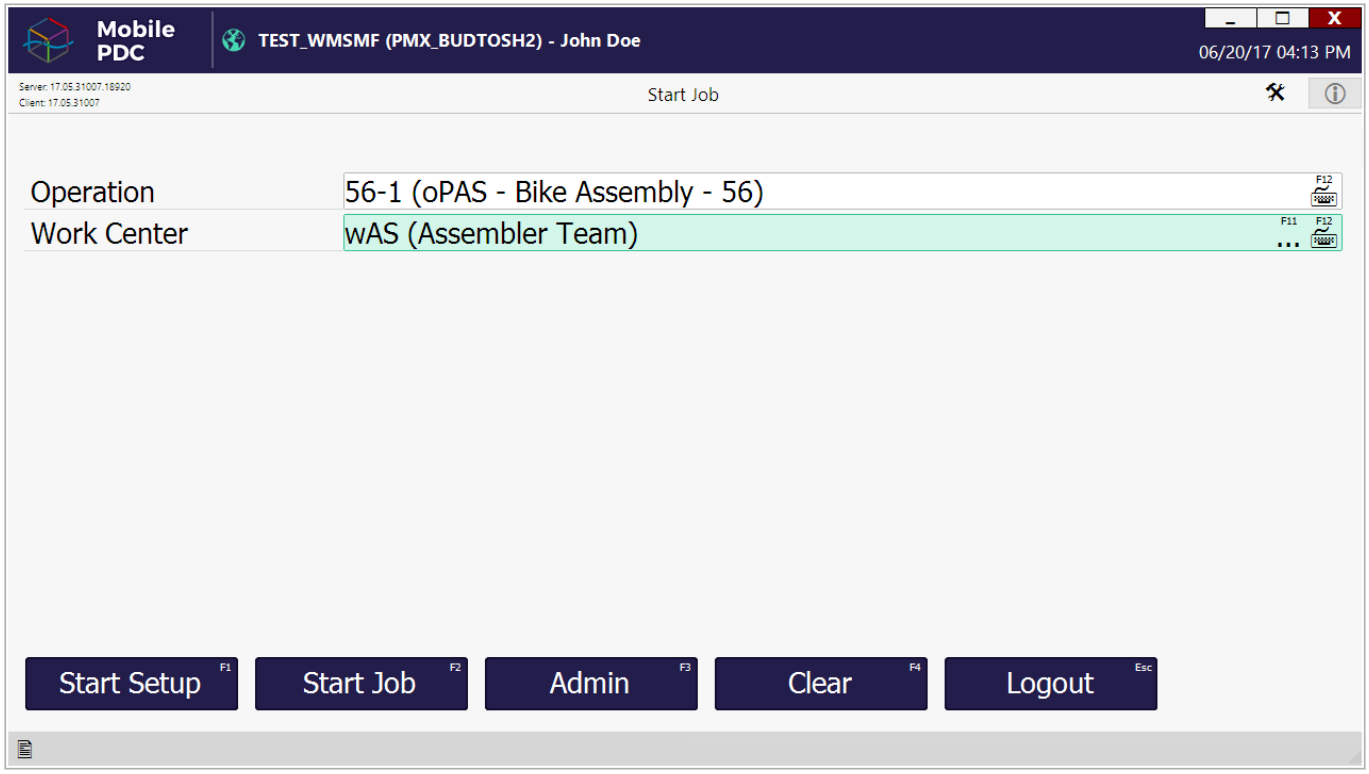
Open the Customization Fields user table and add a new record for an existing field. Set the *Visible* value to 'No' then press the 'Update' button.

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:



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*

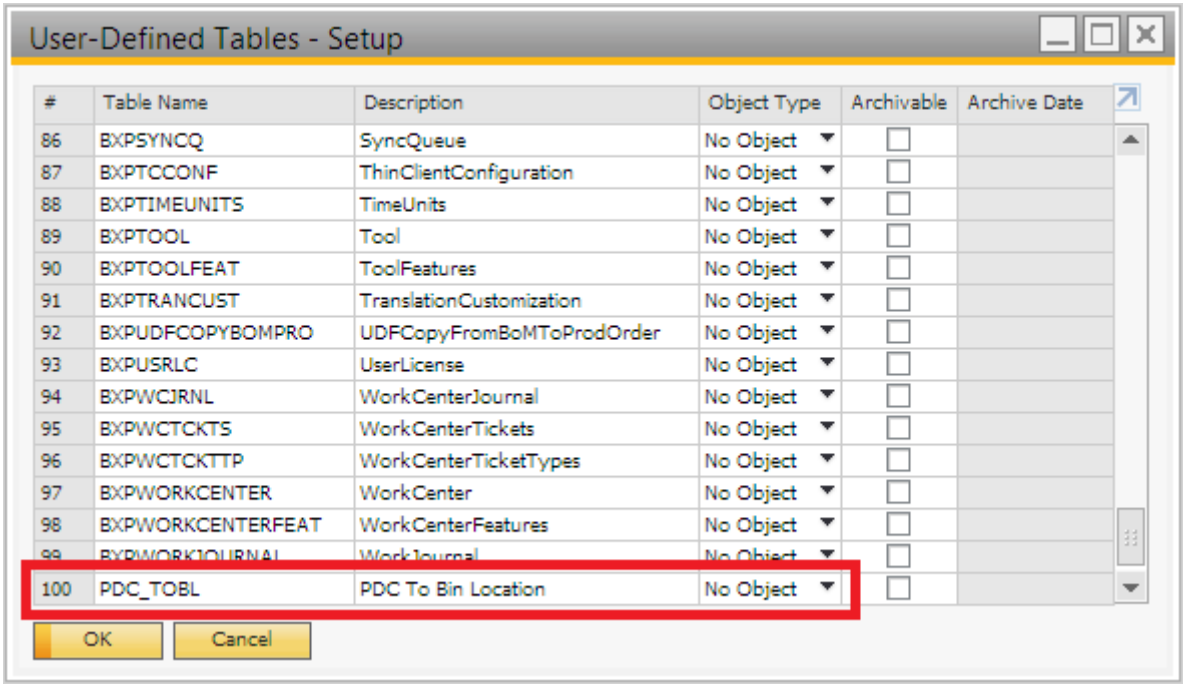
```
IF $[TextEmployeeID] = ''
SELECT 'Scan Employee ID' AS 'Message$', 'E' AS 'MessageType$'
```

For more information about the supported message types please see: Supported message types

6. Save the Destination Bin Location for the employee

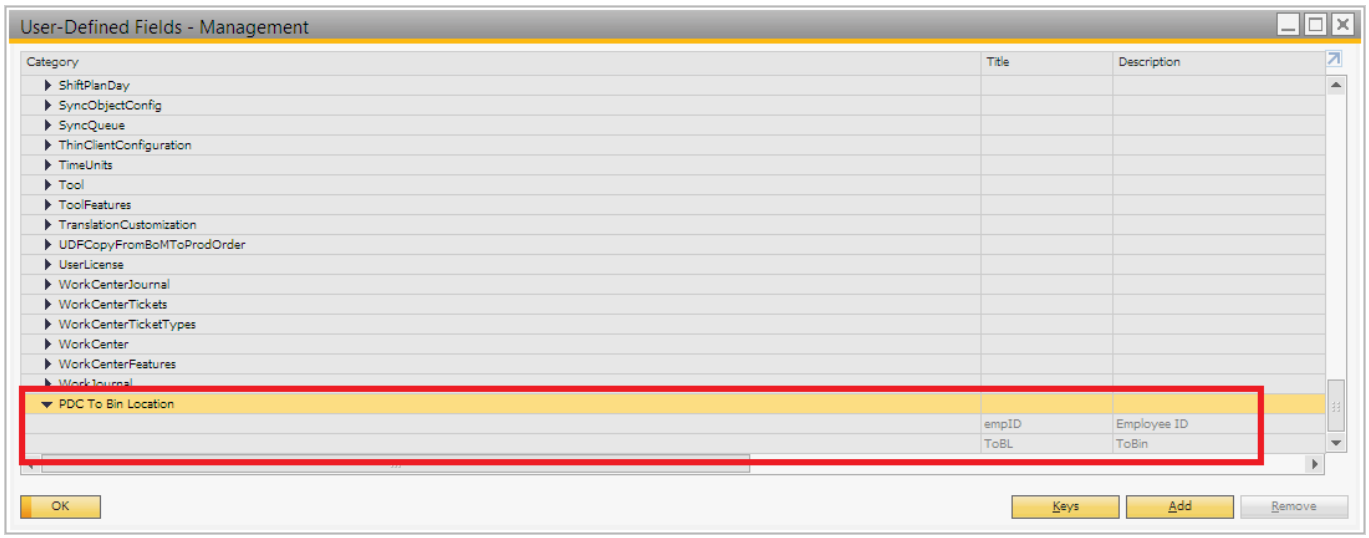
This customization will save the last added bin location on the Products screen. When this screen next loads to the employee, the Bin Location code will be automatically filled. This way if the employee uses the same destination bin location, (s)he does not have to re-enter it.

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

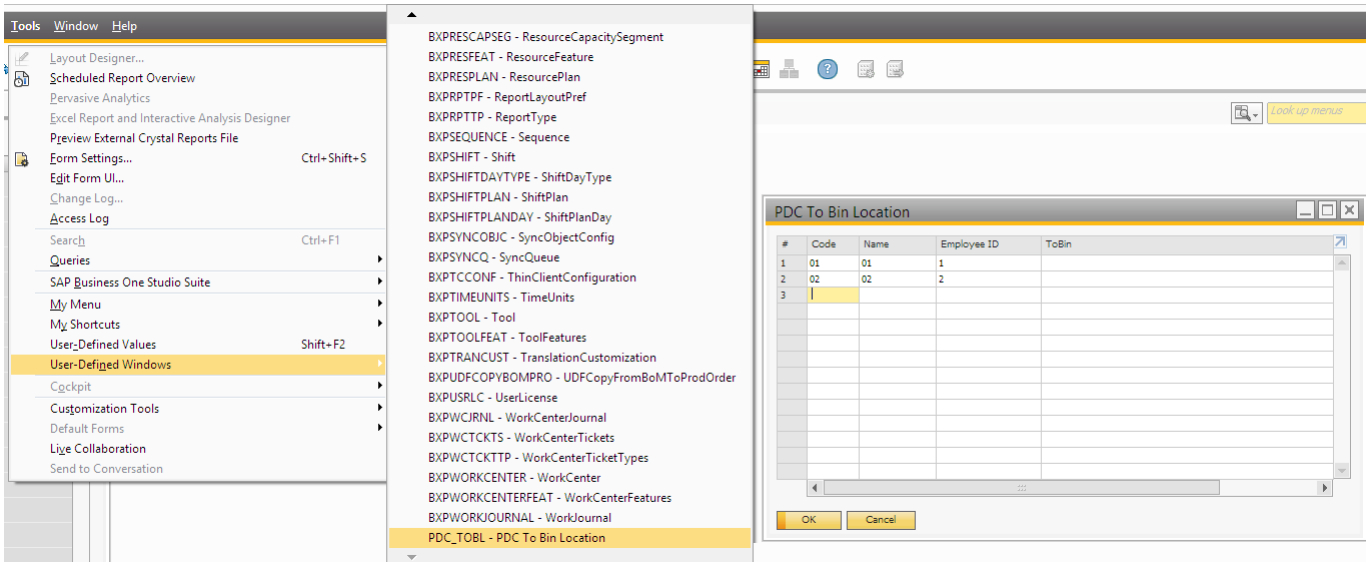


Add the following fields to the user table: [Tools>Customization Tools>User Defined Fields-Management]

- 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'
```

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

Permanent link:
https://wiki.produmex.name/doku.php?id=implementation:manufacturing:customizationexamples

Last update: **2021/04/20 09:24**