# **2. Customization Examples**

## 2.1. Goods Receipt PO

#### **Helpful Tips and Resources**

Click the link below to visit our Article site, where you will find examples and useful information. We are continuously adding new articles featuring the most common customizations.

**Produmex Scan Articles: Customization** 

## 2.1.1. Set (user) fields in GR PO

The customization makes possible to set additional fields (SAP or user fields) in the Goods Receipt PO document from.

Go to GR PO Lines screen, and open Customization window. Here you can see the name of this screen: *GoodsReceiptPOLinesScreen* 

#### ×

To add new fields, open the CustomizationFields User-Defined window in SAP Business One. Use the recommended parameters as below:

Screen	Module	Label	<b>Field Name</b>	Read Only
GoodsReceiptPOLinesScreen	BXMobileWH9	Del.Date	DELDATE	YES

Now you are able to load data into this new field. To do it, you have to use the *BXMobileWH9\_GoodsReceiptPOLinesScreen\_Load* user query.

💐 GRPO	# ℃	( <b>@</b> X	🖳 Customization – 🗆	×
C sh.Ord. Item Filter UoM Quantity Del.Date	10 * Mega Supplier LL pallet Add	C  I Upd	Screen: GoodsReceiptPOLinesScreen \$[ButtonExtraData] = E \$[ButtonCustomizationDebug] = C \$[ButtonUpdate] = Upd \$[ButtonAdd] = Add \$[TextQuantity] = \$[TextUoM] = pallet \$[TextPurchaseOrder] = 10 * Mega Supplier LLC	
#1 A100 Open WH: #2 A100 Open WH:	0 * Normal Paper A1 0 01 0 * Normal Paper A1 0 01 0 / 1	pallet 00 pallet	S[DELDATE] = S[DELDATE] = S[DataRepeater.OlRowStatus] = Open S[DataRepeater.SelectedUIRowStatus] = Open S[DataRepeater.UIWarehouse] = WH: 01 S[DataRepeater.SelectedUIWarehouse] = WH: 01 S[DataRepeater.UIItemCode] = A1000 * Normal Paper A1 000##A1000 * Normal Paper A1 000 S[DataRepeater.SelectedUIItemCode] = A1000 * Normal Paper A1 000 S[DataRepeater.SelectedUIItemCode] = A1000 * Normal Paper A1 000 S[DataRepeater.SelectedUIItemCode] = #1###2 S[DataRepeater.SelectedUILineNumber] = #1 S[DataRepeater.UIQuantityDetails] = 0 / 1 pallet##0 / 1 pallet	~
Done			Close	

If you want to select the delivery date of the purchase order, you will need the purchase order number. You can check the field name on customization screen. In the example: TextPurchaseOrder = 10 \* Mega Supplier LLC

You have to use \$[TextPurchaseOrder] field and you have to split the string in the query. If you want to read the data from the field, then you have to use \$ character.

```
SELECT DocDueDate as [DELDATE] FROM OPOR
WHERE DocNum = SUBSTRING( $[TextPurchaseOrder], 0, CHARINDEX('*',
$[TextPurchaseOrder], 1) - 1)
```



## 2.1.2. Set Freight

It is possible to set the Freight costs/lines in the created Goods Receipt PO with user query. Please see

## 2.1 Set Freight in Delivery document for the example query.

UserQuery: bx_mobile_wh9_document_additionalexpenses (Category: BXMobileWH9)							
Parameters	Results						
[%1] - employeeID	Zero, one or more rows for the Freight						
[%2] - TerminalID	charges to be created. The result columns						
[%3] - Head Code from Mobile Transaction table	can have the names:						
(@BXPLMSMOBTHD.Code)	* BO_LineTotal = the LineTotal field, amount						
[%4] - grouped Head Codes (string, list), separated	of money (mandatory)						
with # characters, only applicable for grouped	* BO_ExpenseCode = Expense Code (integer)						
Purchase Order/APReserve invoice $\rightarrow$ Goods Receipt PO	(mandatory)						

## 2.1.3. Batch number generation

On some screens (e.g. Goods Receipt PO, Receipt from Production) the system can automatically generate batch numbers when entering the received quantities and bin locations. The batch number generation logic can be defined in a user query.

#### UserQuery: bx\_mobile\_wh9\_get\_new\_batchnumber (Category: BXMobileWH9)

Parameters	
[%1]: employee ID (int)	
[%2]: terminal ID (nvarchar)	Results
[%3]: base document type (int)	The user query should return the batch number in a column called
[%4]: base document entry (int)	BXBATNUM (type of the field should be NVARCHAR)
[%5]: base document line (int)	
[%6]: item code (nvarchar)	

Note: The standard *click* and *click\_after* events cannot use the parameters of the bx\_mobile\_wh9\_get\_new\_batchnumber query.

Example:

The following query generates a batch number combining the supplier code and the current date:

SELECT T0.[CardCode] + '-' + CONVERT(varchar, GETDATE(), 112) AS 'BXBATNUM'
FROM OPOR T0 WHERE T0.[DocEntry] = [%4]

## 2.1.4. Serial number generation

In the Goods Receipt PO process, the system can automatically generate serial numbers when entering the received stocks. The serial number generation logic needs to be defined in a user query.

UserQuery: bx_mobile_wh9_get_new_serialnumber (Category: BXMobileWH9)							
Parameters							
[%1]: employee ID (int)	Results						
[%2]: terminal ID (nvarchar)	The user query should return as single serial number in a column						
[%3]: base document type (int)	called BXSERNUM.						
[%4]: base document entry (int)	<b>Note:</b> If the query returns more than one row, the first row's value						
[%5]: base document line (int)	will be used.						
[%6]: item code (nvarchar)							

## 2.1.5. Expiration date at GRPO/Inventory

The following fields can be used at BatchScreen or SerialScreen:

```
BATCH_EXPIRATION_DATE = "#ExpirationDate";
BATCH_MANUFACTURING_DATE = "#ManufacturingDate";
BATCH_ATTRIBUTE1 = "#Attribute1";
BATCH_ATTRIBUTE2 = "#Attribute2";
BATCH_DETAILS = "#Details";
```

```
SERIAL_EXPIRATION_DATE = "#SerialExpirationDate";
SERIAL_MANUFACTURING_DATE = "#SerialManufacturingDate";
SERIAL_ATTRIBUTE1 = "#SerialAttribute1";
SERIAL_ATTRIBUTE2 = "#SerialAttribute2";
SERIAL_DETAILS = "#SerialDetails";
```

These fields can be added on the CustomizationFields table.

Example for batches:

Field Name	Label	Screen
#ExpirationDate	BBD	ReceiptFromProductionQuantitiesBatchScreen
#ManufacturingDate	ManufacturingDate	ReceiptFromProductionQuantitiesBatchScreen
#ExpirationDate	BBD	GoodsReceiptPOQuantitiesBatchScreen

## 2.1.6. Generate GRPO from Draft

Note: This customization is compatible from SAP Business One version 2405 or later versions. It will not function on earlier versions!

With this customization GRPOs can be created from draft. When starting the Goods Receipt Purchase Order function on the scanner and there is a draft GRPO registered in SBO for the given warehouse, the draft document is listed among the Purchase Orders.

The 'Post' button is immediately active for draft documents when selected and the 'Receive' button is hidden/not available. When the 'Post' button is pressed while a GRPO draft is selected, the GRPO is created from the GRPO draft with the exact same stock parameters as the GRPO draft.

×

Note: DR = draft GRPO (new), PO = Purchase Order, RI = A/P Reserve Invoice

#### UserQuery: bx\_mobile\_wh9\_goodsreceiptpo\_query\_custom

```
SELECT
   Distinct "DocEntry", "DocType", ItemCode
FROM (
```

```
SELECT "OPOR"."DocEntry", 22 as "DocType", "CardCode", "DocNum",
"DocDueDate", "POR1"."ItemCode" FROM "POR1" LEFT JOIN "OPOR" ON
"POR1"."DocEntry" = "OPOR"."DocEntry" WHERE "DocStatus" = 'O'
   UNION
   SELECT "OPCH"."DocEntry", 18 as "DocType", "CardCode", "DocNum",
"DocDueDate", "PCH1"."ItemCode" FROM "PCH1" LEFT JOIN "OPCH" ON
"PCH1"."DocEntry" = "OPCH"."DocEntry" WHERE "DocStatus" = '0'
   UNION
   SELECT "ODRF"."DocEntry", 20 as "DocType", "CardCode", "DocNum",
"DocDueDate", "DRF1"."ItemCode" FROM "DRF1" LEFT JOIN "ODRF" ON
"DRF1"."DocEntry" = "ODRF"."DocEntry" WHERE "ODRF"."ObjType" = '20' AND
"DocStatus" = '0' AND "ODRF"."WddStatus" in ('Y','')
    ) as DOCUMEND
WHERE
    (CASE WHEN isnull($[ItemCode], '') = '' THEN 1 ELSE CASE WHEN
"DOCUMEND"."ItemCode" = $[ItemCode] THEN 1 ELSE 0 END END) = 1
   AND (CASE WHEN isnull($[DocNum], '') = '' THEN 1 ELSE CASE WHEN
"DOCUMEND"."DocNum" = $[DocNum] THEN 1 ELSE 0 END END) = 1
    AND (CASE WHEN isnull($[CardCode], '') = '' THEN 1 ELSE CASE WHEN
"DOCUMEND"."CardCode" = $[CardCode] THEN 1 ELSE 0 END END) = 1
    AND (CASE WHEN isnull($[DueDate], '') = '' THEN 1 ELSE CASE WHEN
"DOCUMEND"."DocDueDate" = $[DueDate] THEN 1 ELSE 0 END END) = 1
```

Implementation:

Add the query in the Query Manager in SBO

Query Manag	jer
Query Name	
bx_mobile_wh9	<pre>Jgoodsreceiptpo_query_custom</pre>
Query Categor	γ
General	Manage Categories
System	
▼ General	
bx_mobile_v	wh9_goodsreceiptpo_query_custom
Scan	bx_mobile_wh9_goodsreceiptpo_query_custom
	SELECT "DocEntry", 22 AS "DocType" FROM "OPOR" WHERE "DocStatus" = 'O' UNION
	SELECT "DocEntry", 18 AS "DocType" FROM "OPCH" WHERE "DocStatus" = 'O' UNION
	SELECT "DocEntry", 20 AS "DocType" FROM "ODRF" WHERE "ObjType" = '20' AND "DocStatus" = '0'

## 2.2. Sales Orders

## 2.2.1. Set Freight in Delivery document

By default the Freight in the created Delivery document is 0. It is possible to customize Produmex Scan so freight lines are added with a custom freight calculation algorithm. This algorithm receives a parameter which allows to query the items, quantities, base documents which will be used to create the new Delivery Document.

UserQuery: bx_mobile_wh9_document_additionalexpenses (Category: BXMobileWH9)						
Parameters [%1] - employeeID [%3] - Head Code from Mobile Transaction table (@BXPLMSMOBTHD.Code) [%4] - grouped Head Codes (string, list), separated with # characters, only applicable for grouped Purchase Order/APReserve invoice → Goods Receipt PO	<b>Results</b> Zero, one or more rows for the Freight charges to be created. The result columns can have the names: <i>BO_LineTotal</i> = the LineTotal field, amount of money (mandatory) <i>BO_ExpenseCode</i> = Expense Code (integer) (mandatory)					

Example:

A combined query to copy expenses from Sales Orders and Purchase Orders. Query name: *bx\_mobile\_wh9\_document\_additionalexpenses* 

```
declare @salesOrderDocEntry int
SELECT @salesOrderDocEntry = U BXPBsDcE
FROM [@BXPLMSMOBTLN]
WHERE U BXPHdCd = [%3] AND U BXPBsDcT = 17 AND U BXPDocTy = 15
-- 15-delivery, 17-sales order
IF @salesOrderDocEntry > 0
BEGIN
SELECT
    ExpnsCode as B0_ExpenseCode,
    LineTotal as BO LineTotal,
    17 as BO BaseDocType,
    @salesOrderDocEntry as BO BaseDocEntry,
    LineNum as BO BaseDocLine
FROM RDR3
WHERE DocEntry = @salesOrderDocEntry AND Status <> 'C'
END
declare @purchaseOrderDocEntry int
SELECT @purchaseOrderDocEntry = U BXPBsDcE
FROM [@BXPLMSMOBTLN]
WHERE U BXPHdCd = [%3] AND U BXPBsDcT = 22 AND U BXPDocTy = 20
-- 22-Purchase order, 20-Goods Receipt PO
IF @purchaseOrderDocEntry > 0
BEGIN
SELECT
    ExpnsCode as BO ExpenseCode,
    LineTotal as BO LineTotal,
    22 as BO BaseDocType,
    @purchaseOrderDocEntry as BO_BaseDocEntry,
    LineNum as BO BaseDocLine
FROM POR3
WHERE DocEntry = @purchaseOrderDocEntry AND Status <> 'C'
END
```

HANA version

```
CREATE PROCEDURE PMXSCAN ADDITIONALEXPENSES (
    IN
    pHeadCode NVARCHAR(5000)
)
LANGUAGE SQLSCRIPT
AS
BEGIN
    declare purchaseOrderDocEntry int;
    declare salesOrderDocEntry int;
    SELECT SUM("U BXPBsDcE") into salesOrderDocEntry FROM
    (
        SELECT 0 "U BXPBsDcE" FROM DUMMY
        UNION
        SELECT "U BXPBsDcE"
        FROM "@BXPLMSMOBTLN"
        WHERE "U_BXPHdCd" = pHeadCode AND "U_BXPBsDcT" = 17 AND "U_BXPDocTy"
= 15
    );
    -- 15-delivery, 17-sales order
    IF salesOrderDocEntry > 0 THEN
        SELECT
            "ExpnsCode" as "B0 ExpenseCode",
            "LineTotal" as "BO_LineTotal",
            17 as "BO BaseDocType",
            salesOrderDocEntry as "BO BaseDocEntry",
            "LineNum" as "BO BaseDocLine"
        FROM "RDR3"
        WHERE "DocEntry" = salesOrderDocEntry AND "Status" <> 'C';
    END IF;
    SELECT SUM("U BXPBsDcE") into purchaseOrderDocEntry FROM
    (
        SELECT 0 "U BXPBsDcE" FROM DUMMY
        UNION
        SELECT "U_BXPBsDcE"
        FROM "@BXPLMSMOBTLN"
        WHERE "U BXPHdCd" = pHeadCode AND "U BXPBsDcT" = 22 AND "U BXPDocTy"
= 20
    );
    -- 22-Purchase order, 20-Goods Receipt PO
    IF purchaseOrderDocEntry > 0 THEN
        SELECT
            "ExpnsCode" as "BO ExpenseCode",
            "LineTotal" as "BO LineTotal",
            22 as "BO BaseDocType",
            purchaseOrderDocEntry as "BO BaseDocEntry",
            "LineNum" as "BO BaseDocLine"
        FROM "POR3"
        WHERE "DocEntry" = purchaseOrderDocEntry AND "Status" <> 'C';
    END IF;
```

```
END;
```

```
SAP Query name:
bx_mobile_wh9_document_additionalexpenses
content:
```

call "PMXSCAN\_ADDITIONALEXPENSES"( [%3] );

This example user query looks at the Mobile Transaction (lines) table, filters for Base Document Type = Sales Order, finds the base Sales Order Document DocEntry. It then retrieves the freight expense records related to the Sales Order (from table RDR3), and extracts Expense Code and Line Total, so it essentially copies all the freight charges from the Sales Order without any calculation.

## 2.2.2. Packing ID

It is possible to record packaging ID during picking. The package ID screen can be added to the picking lines screen.

Example:

<b>Field Name</b>	Label	Screen
#Packageld	Package ID	PickingLinesPickNormalScreen
#Packageld	Package ID	PickingLinesPickBatchScreen
#Packageld	Package ID	PickingLinesPickSerialScreen

💐 Picking		•••	‰≮	Ξ×
Pick List	57 * 06/04	/18		
Customer	PCO1234	5*Pa	per Com	pany
ltem	A1000			
Open Qty	5 pcs			
Rec. Bin	01-RECE	VING	-BIN-	
Bin Loc.	01-RECEI	VING-	BIN-LOC/	ATION
UoM	pcs			
Quantity	5		pcs	
Package ID	10001			
Find Stock	s			
Pos	t		Back	

The Package ID is automatically filled with the last Package ID value that was added for the pick list. Both numbers and letters are supported in this field.

The package ID is stored on the MobilePickingData (BXPLMSMOBPICK) user table.

	Code	Name	Batch Number	Bin Code	Is Failed	Is Synchronized	Package Id	Item Code	Picked Quantity	Pick List Abs Entry	Pick List Line Numbe
	01797507	01797507		01-RECEIVING-BIN-LOCATION	No 🔻	Yes	10001	A1000	5	57	
	01797508	01797508	BN72510	01-SYSTEM-BIN-LOCATION	No 🔻	Yes	10002	B1000	5	57	1
	01797509	01797509		01-SYSTEM-BIN-LOCATION	No 🔻	Yes	10003	S1000	1	57	2
	01797510	01797510		01-SYSTEM-BIN-LOCATION	No 🔻	Yes	10003	S1000	1	57	2
	01797511	01797511		01-SYSTEM-BIN-LOCATION	No 🔻	Yes	10003	S1000	1	57	2
	01797512	01797512		01-SYSTEM-BIN-LOCATION	No 🔻	Yes	10003	S1000	1	57	2
7	01797513	01797513		01-SYSTEM-BIN-LOCATION	No 🔻	Yes	10003	S1000	1	57	2
8					No 🔻	No					
•											
0											
1											
2											
13											
14											
15											
16											
17											
18											
19											
20											
	4										

## 2.3. Pick Lists

## 2.3.1. Set Freight in Delivery

Please see: 2.1 Set Freight in Delivery document

## 2.3.2. Set (user) fields in Delivery

It is possible to set the values of the Delivery Document which are created from the Produmex Scan Pick List screen.

To set a field, you have to add a customization field to the Pick List Lines screen. The customization field name must be:

- **BO\_U\_XXField1** to set Delivery Document U\_XXField1 custom field
- **BO\_FieldName** to set Delivery Document SAP internal field by API name. In this case, the field name must match the DI API name. Example: BO\_JournalMemo

In the Customization Fields user table, it is important that the Field Name is  $BO_U$ \*and it must match the UDF name (with a U\_Prefix), eg.  $BO_U$ Test1  $\rightarrow$  Test1 userdefined field. The Label can be anything.

#### Example:

Screen	Module	Label	Field	Name
PickingLinesScreen	BXMobileWH9	Test 1	BO_U	Test1
PickingLinesScreen	BXMobileWH9	Test 2	BO_U	Test2

💐 Picking	#°‰∢ œ×	🗄 Customization 🗕 🗆 😒	×
C : List Customer Bin/Item	8 * 08/10/17 PPLC199 * Pulp&Paper	Screen: PickingLinesScreen \$[ButtonExtraData] = E \$[ButtonCustomizationDebug] = C \$[BO_U_Test2] = bbbb \$[BO_U_Test1] = aaaa	
Test 1 Test 2	aaaa bbbb	S[DettonClear Line] - Clear Line S[TextQuantity] = 1 S[ButtonDeliver] = Deliver S[TextBinCode] -	
#1 A100 Picked 01-S SO	00 * Normal Paper A1 000 ST-P0011 1 / 1 box	S[TextCustomer] = PPLC199 * Pulp&Paper S[TextPickList] = 8 * 08/10/17 S[DataRepeater.UIBaseType] = SO##SO##SO S[DataRepeater.SelectedUIBaseType] = SO	
#2 B100 Picked 01-S SO B111	00 * B1 batch paper ST-P0011 70803 10 / 10	<pre>\$[DataRepeater.UlBatchNumber] = B1170803## \$[DataRepeater.SelectedUlBatchNumber] = \$[DataRepeater.UlBinCode] = 01-ST-P0011##01-ST-P0011##01-SYSTEM-BIN-LOCATION \$[DataRepeater.SelectedUlBinCode] = 01-ST-P0011 \$[DataRepeater.Ultem] = A1000 * Normal Paper A1 000##B1000 * B1 batch paper##\$1000 * \$1</pre>	
Clear Line	e Deliver	serialized paper	~
Pick	Back	Close	

When pressing the Delivery button on the Pick List lines screen, these field values will be used in the newly created Delivery Document.

## 2.3.3. Pick List screen - customize list

With the default settings, only open pick lists that have not been started by anyone are displayed, but the original program logic can be overridden by customization.

UserQuery: bx_mobile_wh9_picklists	s_query_custom (Category: BXMobileWH9)
Parameters \$[AbsEntry] \$[CardCode] \$[EmployeeNo] - logged in employeeID \$[ItemCode] \$[PickDate] \$[WarehouseCode]	<b>Results</b> A table with multiple rows with a single column (integer) with PickList AbsEntry values. (OPKL.AbsEntry)

Example:

With the example user query the list of picklists is filtered down to picklist assigned to the employee. We used the 'Picker' field on the Pick list to assign the employee to the pick list. Please note: The Picker field must be in the 'EirstName LastName' or 'EirstName MiddleName

*Please note: The Picker field must be in the 'FirstName LastName' or 'FirstName MiddleName LastName' format for the example user query to work.* 

Pic	k List								
Pic	k Number		12		Remarks				
Pic	k Date		08/10/17		]				
Use	er		manager		1				
Pic	ker		John Doe		]				
Sta	itus		Released						
#	Trans. T	Doc. No.	BP Code	Number	UoM Code	UoM Name	Items per Unit	Whse	Whse
1	OR	📫 16	PCO1234	📫 A1000	pcs	pieces	1	📫 01	General \

The example user query name: *bx\_mobile\_wh9\_picklists\_query\_custom* 

```
IF $[AbsEntry] IS NOT NULL
BEGIN
    SELECT AbsEntry
    FROM OPKL WHERE AbsEntry = $[AbsEntry] AND Status <> 'C'
END
ELSE
BEGIN
    SELECT AbsEntry
    FROM OPKL
    WHERE Name = (SELECT firstName + ' ' + ISNULL(middleName + ' ', '') +
lastName
    FROM OHEM
    WHERE empID = $[EmployeeNo]) AND Status <> 'C'
    ORDER BY AbsEntry
END
```

This simple example only filters for employee or PickListNo, but doesn't respect other filters like Customer, Item, DueDate, Warehouse. It also allows the employee to enter a PickListNo and allows him to select that pick list even if he's not assigned for it.

## 2.3.4. Auto fill batch with recommended batch

To automatically populate the Batch field with the recommended batch, add the following query:

Query name: BXMobileWH9\_PickingLinesPickBatchScreen\_Activate

SELECT \$[TextRecBatch] AS [TextBatch]

## 2.3.5. Capture pick list selection events into a separate table

It is possible to capture the date and time when an employee selects a pick list into a separate table.

#### Create the user table

First create the user table for the pick list selection events. Example: PMX\_PLLOG user table

Set the object type to 'No object with Auto.Increment'.

ŧ	Table Name	Description	Object Type		Archivable	Archive Date	2
26	BXPMWGENCON	Produmex PDC General Co	No Object	•			
27	BXPREPORT	ReportModule	No Object	-			
28	BXPRPTPF	ReportLayoutPref	No Object	•			
29	BXPRPTTP	ReportType	No Object	•			
30	BXPSEQUENCE	Sequence	No Object	•			
31	BXPSYNCOBJC	SyncObjectConfig	No Object	•			
32	BXPSYNCQ	SyncQueue	No Object	•			
33	BXPTCCONF	ThinClientConfiguration	No Object	•			
34	BXPTRANCUST	TranslationCustomization	No Object	•			
35	BXPUPDSCRPT	Upgrade Scripts	No Object	•			
36	BXPUSRLC	UserLicense	No Object	•			
37	BXPVERSIONS	Module Versions	No Object	•			
38	PMX_EMPLOG	Employee Log	No Object with Auto. Increment	•			1
39	PMX_PLLOG	Pick List Log	No Object with Auto. Increment	•			
40			No Object	•			

Add the user defined fields to the table. In the example we will add the following fields:

Title	Description	Туре
Date	Date	Date/Time
Time	Time	Numeric
EmpID	Employee ID	Alphanumeric
EmpName	Employee Name	Alphanumeric
PLN	Pick List	Numeric

#### Create the user query

The user query name is: BXMobileWH9\_PickingScreen\_OK\_clicked

SQL

```
INSERT INTO "@PMX_PLLOG" ("Name", "U_Date", "U_Time", "U_EmpID",
"U_EmpName", "U_PLN")
values ('Pick List Entry', cast(getdate() as date),
cast(substring(CONVERT(VARCHAR,GETDATE(),108),1,2) * 100 +
substring(CONVERT(VARCHAR,GETDATE(),108),4,2) as int),
$[Employee.EmployeeID], $[Employee.FirstName]+ ' '+ $[Employee.LastName],
SUBSTRING($[DataRepeater.SelectedUIPickListNo], CHARINDEX('#',
$[DataRepeater.SelectedUIPickListNo]) + 1,
LEN($[DataRepeater.SelectedUIPickListNo]) - CHARINDEX('#',
$[DataRepeater.SelectedUIPickListNo]))
```

HANA

INSERT INTO "@PMX\_PLLOG" ("Name", "U\_Date", "U\_Time", "U\_EmpID", "U\_EmpName", "U\_PLN")

```
values ('Pick List Entry', cast(getdate() as date),
cast(substring(CONVERT(VARCHAR,GETDATE(),108),1,2) * 100 +
substring(CONVERT(VARCHAR,GETDATE(),108),4,2) as int),
$[Employee.EmployeeID], $[Employee.FirstName]+ ' '+ $[Employee.LastName],
SUBSTRING($[DataRepeater.SelectedUIPickListNo],
LOCATE($[DataRepeater.SelectedUIPickListNo], '#') + 1) FROM DUMMY')
```

When the 'Pick' button is pressed, this query inserts the current date to the Date column, the current time to the Time column, the employee ID to the Employee ID column, the first and last name of the employee to the Employee Name column and the pick list number to the Pick List column.

#### ×

## **2.4. Query Stocks**

## 2.4.1. Override list

It is possible to override the list of items on the Query Stocks screen.

This is the same screen that can be opened from Pick List and from other modules in Produmex Scan when pressing the Find Stocks button, so the custom logic is also relevant for those cases.

UserQuery: bx_mobile_wh9_quer	ystocks_query_custom (Category: BXMobileWH9)
	Results
Parameters	A table with multiple rows with specific column names:
\$[Warehouse]	- Warehouse
\$[BinLocation]	- BinLocation
\$[ItemCode]	- ItemCode
\$[BatchNumber]	- ItemName
(\$[] - other user fields from screen)	- ManagedBy (Batch: 10000044, Serial: 10000045, None: -1) - OnHandQuantity (in inventory UoM)

Example:

This custom query returns items sorted by quantity (descending). Please note: This query doesn't filter by batch input parameter, only Warehouse, ItemCode and BinLocation.

Query name: bx\_mobile\_wh9\_querystocks\_query\_custom

```
-- return maximum 20(+20) matches by filters, ordered by quantity descending
-- first select is for bin-activated warehouses
SELECT TOP 20 OIBQ.WhsCode as Warehouse, OBIN.BinCode as BinLocation,
OIBQ.ItemCode, OITM.ItemName,
CASE
WHEN OITM.ManSerNum = 'Y' THEN 10000045
WHEN OITM.ManBtchNum = 'Y' THEN 10000044
```

```
ELSE -1 END as ManagedBy, OIBQ.OnHandQty as OnHandQuantity FROM OIBQ
JOIN OBIN ON (OBIN.AbsEntry = OIBQ.BinAbs)
JOIN OITM ON (OITM.ItemCode = OIBQ.ItemCode)
WHERE (OIBQ.ItemCode = $[ItemCode] OR $[ItemCode] = '')
   AND (OIBQ.WhsCode = $[Warehouse] OR $[Warehouse] = '')
   AND (OBIN.BinCode = $[BinLocation] OR $[BinLocation] = '')
   AND OnHandOty > 0
UNION
-- this second select is for non-bin warehouses
SELECT TOP 20 OITW.WhsCode as Warehouse, '' as BinLocation, OITW.ItemCode,
OITM.ItemName,
CASE
   WHEN OITM.ManSerNum = 'Y' THEN 10000045
   WHEN OITM.ManBtchNum = 'Y' THEN 10000044
   ELSE -1 END as ManagedBy, OITW.OnHand as OnHandQuantity FROM OITW
JOIN OITM ON (OITM.ItemCode = OITW.ItemCode)
JOIN OWHS ON (OWHS.WhsCode = OITW.WhsCode)
WHERE OWHS.BinActivat = 'N'
   AND (OITW.ItemCode = $[ItemCode] OR $[ItemCode] = '')
   AND (OITW.WhsCode = $[Warehouse] OR $[Warehouse] = '')
-- order by quantity descending
ORDER BY OnHandQuantity DESC
```

## 2.5. General for multiple processes

## 2.5.1. Creating documents as drafts

It is possible to control whether the documents should be created as drafts or as real documents when posted from Produmex Scan for the following documents:

Document	Doc. type
Delivery	15
Sales Order (by BN Create SO function)	17
Goods Receipt PO	20
Goods Issue	60

The controlling logic must be defined with a user query.

UserQuery: bx_mobile_wh9_docume	lserQuery: bx_mobile_wh9_document_creation_type (Category: BXMobileWH9)			
	Results			
Parameters	The user query should return the result in a column called			
[%1]: employee ID (int)	BXDOCTYP. The result must be an integer, and the			
[%3]: mobile transaction head code	following values are supported:			
(nvarchar)	- 0: real document			
	- 1: draft			

For example, with the following logic, all goods receipt PO documents (doc. type = 20) will be created

as drafts, while the other documents are created as real documents:

MS SQL

```
SELECT CASE T0.[U_BXPDocTy]
WHEN 20 THEN 1
ELSE 0 END as 'BXD0CTYP'
FROM [dbo].[@BXPLMSM0BTHD] T0 WHERE T0.[Code] = [%3]
```

HANA

```
SELECT
case
T0."U_BXPDocTy" when 20 then 1
else 0
end as "BXDOCTYP"
FROM "@BXPLMSMOBTHD" T0 WHERE T0."Code" = [%3]
```

#### 2.5.2. Special field for series numbering

A user field for series numbering can be added to screens where a Post event starts. Field name: BO\_Series

Example: Add a user field for numbering series to the GR PO screen Add the following record to the Customization Fields user table:

Screen	Module	Label	Field Name	Read Only
GoodsReceiptPOPostSelectionScreen	BXMobileWH9	Series numbering	BO_Series	No

#### 2.5.3.Capture login events into a separate table

It is possible to capture the date and time of the employee login and logout events in a separate user table.

#### Create the user table

First create the user table for the log in and log out events. Example: PMX\_EMPLOG user table

Set the object type to 'No object with Auto.Increment'.

#	Table Name	Description	Object Type		Archivable	Archive Date	2
26	BXPMWGENCON	Produmex PDC General Co	No Object	٠			4
27	BXPREPORT	ReportModule	No Object	•			
28	BXPRPTPF	ReportLayoutPref	No Object	•			
29	BXPRPTTP	ReportType	No Object	•			
30	BXPSEQUENCE	Sequence	No Object	•			
31	BXPSYNCOBJC	SyncObjectConfig	No Object	•			
32	BXPSYNCQ	SyncQueue	No Object	•			
33	BXPTCCONF	ThinClientConfiguration	No Object	•			
34	BXPTRANCUST	TranslationCustomization	No Object	•			
35	BXPUPDSCRPT	Upgrade Scripts	No Object	٠			
36	BXPUSRLC	UserLicense	No Object	•			
37	BXPVERSIONS	Module Versions	No Object	٠			
38	PMX_EMPLOG	Employee Log	No Object with Auto, Increment	•			
39	PMX_PLLOG	Pick List Log	No Object with Auto, Increment	٠			
40			No Object	•			

Add the user defined fields to the table. In the example we will add the following fields:

Title	Description	Туре
Date	Date	Date/Time
Time	Time	Numeric
EmpID	Employee ID	Alphanumeric
EmpName	Employee Name	Alphanumeric

#### Create the user query

The user query name for the log in is: BXMobileWH9\_LoginScreen\_OK\_clicked

SQL

```
INSERT INTO "@PMX_EMPLOG" ("Name", "U_Date", "U_Time", "U_EmpID",
"U_EmpName")
values ('Login', cast(getdate() as date),
cast(substring(CONVERT(VARCHAR,GETDATE(),108),1,2) * 100 +
substring(CONVERT(VARCHAR,GETDATE(),108),4,2) as int), $[TextUser],
$[TextUserName])
```

HANA

```
INSERT INTO "@PMX_EMPLOG" ("Name", "U_Date", "U_Time", "U_EmpID",
"U_EmpName")
values ('Login', cast(current_timestamp as date), T0_INT( T0_VARCHAR(
CURRENT_TIMESTAMP, 'HH24MI' ) ), $[TextUser], $[TextUserName])
```

This query inserts the current date to the Date column, the current time to the Time column, the employee ID to the Employee ID column, the first and last name of the employee to the Employee

Name column and the event name in the Name column.

Emp	oloyee Lo	g				_	. 🗆 🗙
#	Code	Name	Date	Time	EmpID	EmpName	7
1	1	Login	02/27/18	1227	1	John Doe	-
2	2	Login	02/27/18	1228	3	Tess Smith	
3							
		_					
							_
		_					_
		_					_
	ОК	Cancel					

## 2.6. Special customization

You can use this special customization to manipulate the loaded data.

## 2.6.1. Special customization queries

#### 2.6.1.1. Query Stocks

User query: bx_mobile_wh9_querystocks_query_custom			
	Query fields		
	Warehouse,		
Parameters	BinLocation,		
\$[ltemCode],	ltemCode,		
\$[Warehouse]	ltemName,		
	ManagedBy,		
	OnHandQuantity		

This example will order the result by batch number.

```
SELECT TOP 20 0IBQ.WhsCode as Warehouse, OBIN.BinCode as BinLocation,
OIBQ.ItemCode, OITM.ItemName,
CASE
    WHEN 0ITM.ManSerNum = 'Y' THEN 10000045
    WHEN 0ITM.ManBtchNum = 'Y' THEN 10000044
    ELSE -1 END as ManagedBy, OIBQ.OnHandQty as OnHandQuantity
FROM 0IBQ
    JOIN OBIN ON (OBIN.AbsEntry = OIBQ.BinAbs)
    JOIN 0ITM ON (OITM.ItemCode = OIBQ.ItemCode)
```

```
JOIN OBBQ ON (OBBQ.BinAbs = OIBQ.BinAbs)
JOIN OBTN ON (OBBQ.SnBMDAbs = OBTN.AbsEntry)
WHERE
  (OIBQ.ItemCode = $[ItemCode] OR $[ItemCode] = '')
  AND (OIBQ.WhsCode = $[Warehouse] OR $[Warehouse] = '')
  AND (OBBQ.ItemCode = $[ItemCode] OR $[ItemCode] = '')
  AND (OBBQ.WhsCode = $[Warehouse] OR $[Warehouse] = '')
  AND (OBBQ.OnHandQty > 0
  AND OBBQ.OnHandQty > 0
  ORDER BY OBTN.DistNumber
```

#### 2.6.1.2. Picklist

User query: bx_mobile_wh9_picklists_query_custom				
Parameters \$[AbsEntry], \$[CardCode], \$[EmployeeNo], \$[ItemCode], \$[PickDate], \$[WarehouseCode], \$[BinLocationCode]	Query fields AbsEntry			

On the pick list screen, you can use custom fields in the custom query. Example:

Add a new custom field 'CFEmpID' for employee input on the CustomizationFields table.

<b>Field Name</b>	Label	Screen
CFEmpID	EmpID	PickingScreen

After the field is added it can be used in a custom query. Query name: bx\_mobile\_wh9\_picklist\_query\_custom

```
IF ($[CFEmpID]='')
BEGIN
    SELECT AbsEntry FROM OPKL WHERE Status <> 'C'
    ORDER BY AbsEntry desc
END
ELSE
BEGIN
    SELECT AbsEntry FROM OPKL WHERE Status <> 'C' AND U_BXPEmpID=$[CFEmpID]
    ORDER BY AbsEntry desc
END
```

#### 2.6.1.3. Sales order lines

#### User query: bx\_mobile\_wh9\_salesorderlines\_query\_custom

Parameters	Query fields
\$[DocEntry]	LineNum,
	BINCODE

#### 2.6.1.4. Sales issue

User query: bx_mobile_wh9_salesissue_query_custom				
Parameters \$[DocNum], \$[CardCode], \$[DueDate], \$[ItemCode], \$[EmployeeNo]	<b>Query fields</b> DocEntry, DocType (17:Sales Order; 13:A/R Reserve invoice)			

#### 2.6.1.5. Goods Receipt PO

User query: bx_mobile_wh9_goodsreceiptpo_query_custom				
Parameters \$[DocNum], \$[CardCode], \$[DueDate], \$[ItemCode], \$[EmployeeNo]	<b>Query fields</b> DocEntry, DocType (18:A/P Reservere invoice; 22: Purchase Order)			

#### 2.6.1.6. Stock Transfer Request

User query: bx_mobile_wh9_stocktransferrequest_query_custom				
Parameters				
\$[DocNum]	Query fields			
\$[WarehouseFrom]	DocEntry			
\$[WarehouseTo]	-			

#### 2.6.1.7. Stock Transfer Request Lines

User query: bx_mobile_wh9_stocktransferrequestlines_query_custom			
<b>Parameters</b> \$[DocEntry]	<b>Query fields</b> LineNum IsAvailable		

#### 2.6.1.8. Customize The order of the pick list lines

This query is designed to work with both **Speed Picking** and **Sequential Picking** processes. In both cases, the system will place the picked items at the end of the list without reloading the entire item list. This behavior is intentional and should remain the standard for these picking logics to maintain consistency and efficiency.

Important Note: The "PickEntry" field and the "Order" alias are mandatory and must be used!

User query: bx_mobile_wh9_picklistlines_query_custor				
Parameters	Query fields			
\$[AbsEntry]	PickEntry AS "Order"			

```
IF $[AbsEntry] IS NOT NULL
BEGIN
    SELECT PickEntry AS "Order"
    FROM PKL1 WHERE AbsEntry = $[AbsEntry]
    ORDER BY RelQtty DESC, PickEntry DESC
END
```

### 2.6.2. Button customization

Example: Automatically click on the 'Reload' button after scanning on the GR PO screen.

Query name: BXMobileWH9\_GoodsReceiptPOScreen\_TextDocumentNumber\_validate\_after

```
IF $[TextDocumentNumber] <> ''
BEGIN
SELECT 'ButtonReload' as 'Click$'
END
```

The name of the button can be found in customization assist. A custom button also can be pressed. The two buttons at the bottom of the screen are called 'OK' and 'Option'

#### 2.6.3. Custom message

You can send a message to employee is a custom event.

SELECT 'Information' as "Message\$", 'I' as "MessageType\$"

HANA version:

```
SELECT 'Information' as "Message$", 'I' as "MessageType$" FROM DUMMY
```

See the supported message types here: Supported message types

## 2.7. Other Examples

#### 2.7.1. Populate Bin Location field with default item locaion in GR PO

You have to add the same query content for all item types

Query names:

BXMobileWH9\_GoodsReceiptPOQuantitiesNormalScreen\_Load BXMobileWH9\_GoodsReceiptPOQuantitiesBatchScreen\_Load BXMobileWH9\_GoodsReceiptPOQuantitiesSerialScreen\_Load

```
select
    "OBIN"."BinCode" as "TextBinLocation"
from
    "OITW"
    LEFT JOIN "OBIN" on "OITW"."DftBinAbs" = "OBIN"."AbsEntry"
where
    "OITW"."ItemCode" = $[SelectedPurchaseOrderLine.ItemCode]
    and "OITW"."WhsCode" = $[SelectedPurchaseOrderLine.WarehouseCode]
```

#### 2.7.2. Delete the document number in screen load event in GR PO Screen

You have to add an SAP query with the name of the screen load event Query names:

BXMobileWH9\_GoodsReceiptPOScreen\_Load

select '' as "TextDocumentNumber"

#### 2.7.3. Auto print Delivery in Sales Issue process

It is possible to use the ButtonPost\_click\_after event to trigger the printing. The SAP document is already created in this event. The number of the new Delivery is not available in the customization, so we can only print the last Delivery document that was created by the logged in employee.

You can see the configuration steps below.

- Locate the Delivery note from the "Report and Layout Manager"

Report and Layout Manager						
List <u>S</u> earch		Layouts	Printing Segs			
Asterisk Indicates 🔽 Report 🔽 Layout	#	Layout ID	Layout Name	Туре	Language	7
Administration*	1	DLN20001	Packing List (System)	PLD	English	
Financials*	2	DLN20002	A/R Delivery Note (Item) - CR (US	Crystal Reports	English	
CRM*	3	DLN20003	Packing Slip Blank Stock_USA (System	PLD	English	
Opportunities*	4	DLN20004	Packing List - CR (US) (System)	Crystal Reports	English	
▼ Sales - A/R*	5	DLN20005	Packing Slip Preprinted USA (System)	PLD	English	
Sales Quotation*	6	DLN20007	Delivery Including Batch/SN (System)	PLD	English	
Sales Order*	7	DLN20009	Packing List (EhP) (System)	PLD	English	
▼ Delivery*	8	DLN20011	Packing Slip (EhP) (System)	PLD	English	-
Delivery Note (Service)		4				•
Delivery Note (Items)						
Delivery Note (Document Summary)						
A/R Packaging	ID	ID DLN20002 Descr				
Return Request*	Nan	ne	A/R Delivery Note (Item) -			

- Open it in Crystal Reports Designer.

- Set the database configuration for the report file and save a copy from the report.
- Check the paramters of the report file, we need them in the custom query.



-import the report file into SBO by function "Report and Layout Manager"

```
- Create the custom query
```

- Use the SBO Report ID in column: "PrintLayout\$"
- Set all properties in a new column by addind "Print\_" prefix
- Set the printer name in column: "PrintPrinter\$"
- Send a message about the printing in columns "Message\$" and "MessageType\$"

```
SELECT TOP 1
    'RCRI0013' as "PrintLayout$",
    "DocEntry" as "Print_DocKey@",
    15 as "Print_ObjectId@",
    'Bullzip PDF Printer' as "PrintPrinter$",
    'Delivey is Printed' "Message$", 'I' "MessageType$"
FROM
    "ODLN"
WHERE
    "U_BXPEmpID" = $[Employee.EmployeeID]
ORDER BY "DocEntry" DESC
```

- The name od the query must be: BXMobileWH9\_SalesIssueScreen\_ButtonPost\_click\_after

- Produmex Scan application must be restarted after adding the new customization

- Since the report file contains sub reports you may enable it in the setting below. You can find more information about this setting on the url below

#### Advanced printing configurations

					A		
	User-Defined <u>W</u> indows		•	BOY_SBO_LICDATA - BOY_SBO_LICDATA			
	Coc <u>k</u> pit			•	BXPAUDIT - Audit		
	Customization Too	ols		۲.	BXPCONFIG - Configuration		
Con	figuration						
#	Code 🛛 🖓	Name	Description			Value	
19	BXMPRAO	BXMPRAO	Crystal Reports connection parameters		4		

#### In the Picking process you can use the query name below:

BXMobileWH9\_PickingLinesScreen\_ButtonDeliver\_click\_after

The PickingLineScreen will be closed after createing the Delivery and the event won't be able to use objects from the PickingLineScreen anymore. It is necessary to check the setting below in order to prevent closing the screen.

"Don't close screen after picking delivery"

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

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

Last update: 2025/07/14 09:24

