Customizing serial number management in Produmex Scan

This document provides examples for serial number management customization in Produmex Scan. For more information about the general customization method please see: Customization Technology in Produmex Scan

User-Defined Fields

Goods Receipt

Goods Receipt PO Lines Screen

Add the following record to the Customization Fields user table:

FieldName	Visible	Screen
TextUoM	NO	GoodsReceiptPOLinesScreen

Item code field must be scanned before Add button click
Query name: BXMobileWH9_GoodsReceiptPOLinesScreen_ButtonAdd_click

```
IF $[TextItem] =''
SELECT 'You must scan an item first!' AS 'Message$', 'E' AS 'MessageType$',
'TextItem' AS Click$
```

Item code field must be scanned before Update button click
Query name: BXMobileWH9_GoodsReceiptPOLinesScreen_ButtonUpdate_click

```
IF $[TextItem] =''
SELECT 'Please scan an item!' AS 'Message$', 'E' AS 'MessageType$',
'TextItem' AS Click$
```

Focus on Item field (when returning to this screen)
Query name: BXMobileWH9_GoodsReceiptPOLinesScreen_Activate

```
SELECT 'TextItem' AS Click$
```

Change contents of the list, show bar code instead of WH text
Query name: BXMobileWH9_GoodsReceiptPOLinesScreen_DataRepeater_InternalDataLoad

```
SELECT
SUBSTRING(S.splitdata,1,50) AS [DataRepeater.UIItemCode],
C.CodeBars AS [DataRepeater.UIWarehouse]
FROM dbo.SplitStringForDataRepeater($[DataRepeater.UIItemCode]) S, OITM C
```

```
WHERE C.ItemCode=SUBSTRING(S.splitdata,1,PATINDEX('%*%',S.splitdata)-2)
```

After Item field has been entered write ' ' to quantity and press Add button
Query name: BXMobileWH9_GoodsReceiptPOLinesScreen_TextItem_validate_after

```
IF $[TextItem]<>'' SELECT '' AS TextQuantity, 'ButtonAdd' AS Click$
```

Goods Receipt PO Quantities Serial Screen

Add the following record to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	GoodsReceiptPOQuantitiesSerialScreen

Create a new user table 'SCANEMPBL' with the following fields:

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

Load bin location for employee, fill EanCode with bar code
Query name: BXMobileWH9_GoodsReceiptPOQuantitiesSerialScreen_Load

```
SELECT
(SELECT [@SCANEMPBL].U_ToBL FROM [@SCANEMPBL] WHERE U_empID =
$[Employee.EmployeeID]) AS 'TextBinLocation',
(SELECT CodeBars FROM OITM WHERE
ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode'
```

Save the Bin Location value for next time
Query name: BXMobileWH9_GoodsReceiptPOQuantitiesSerialScreen_TextBinLocation_validate_after

```
IF (SELECT ISNULL(OHEM.U_ToBL,'') FROM OHEM
WHERE empID = $[Employee.EmployeeID]) <> $[TextBinLocation]
UPDATE OHEM SET [U_ToBL]=$[TextBinLocation] WHERE empID =
$[Employee.EmployeeID]
```

After serial number was specified press OK to go back
Query name: BXMobileWH9_GoodsReceiptPOQuantitiesSerialScreen_TextSerial_validate_after

```
IF $[TextSerial]<>'' SELECT 'OK' AS Click$
```

Goods Receipt PO Quantities Normal Screen

Add the following record to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	GoodsReceiptPOQuantitiesNormalScreen
FieldName	**Visible**	**Screen**	
TextUoM	NO	GoodsReceiptPOQuantitiesNormalScreen	

Create a new user table ‘SCANEMPBL’ with the following fields:

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

Save the Bin Location value for next time
Query name: BXMobileWH9_GoodsReceiptPOQuantitiesNormalScreen_TextBinLocation_validate_after

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

Load last location for employee, EAN bar code, set quantity to 1 and focus on quantity
Query name: BXMobileWH9_GoodsReceiptPOQuantitiesNormalScreen_Load

```
SELECT
(SELECT CodeBars FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode',
(SELECT [@SCANEMPBL].U_ToBL FROM [@SCANEMPBL]  WHERE U_empID =
$[Employee.EmployeeID]) AS 'TextBinLocation', '1' AS 'TextQuantity'
,'TextQuantity' AS Click$
```

After the Add button press OK to go back
Query name: BXMobileWH9_GoodsReceiptPOQuantitiesNormalScreen_ButtonAdd_click_after

```
SELECT 'OK' AS Click$
```

Free Goods Receipt

CreateGoodsReceiptPOLinesScreen

Add the following record to the Customization Fields user table:

FieldName	Visible	Screen
TextUoM	NO	GoodsReceiptReceiptLinesScreen

Create a new user table ‘SCANEMPWH’ with the following fields:

- Employee ID (empID)
- To Warehouse (ToWh)

Load saved warehouse for employee, focus on item
Query name: BXMobileWH9_CreateGoodsReceiptPOLinesScreen_Load

```
SELECT
(SELECT [@SCANEMPWH].U_ToWh FROM [@SCANEMPWH] WHERE U_empID =
$[Employee.EmployeeID]) AS 'TextWarehouse',
'TextItem' AS Click$
```

Focus on item (after returning to this screen) Query name: BXMobileWH9_CreateGoodsReceiptPOLinesScreen_Activate

```
SELECT 'TextItem' AS Click$
```

Modify list contents
Query name: BXMobileWH9_CreateGoodsReceiptPOLinesScreen_DataRepeater_InternalDataLoad

```
-- Modify list contents:
-- [DataRepeater.UIItem] Item code + name
-- [DataRepeater.UIWarehouse] Bar code (EAN)
SELECT SUBSTRING(S.splitdata,7,50) AS [DataRepeater.UIItem], C.CodeBars AS
[DataRepeater.UIWarehouse]
FROM dbo.SplitStringForDataRepeater($[DataRepeater.UIItem]) S, OITM C
WHERE C.ItemCode=SUBSTRING(S.splitdata,7,PATINDEX('%*%',S.splitdata)-8)
```

Save warehouse for employee
Query name: BXMobileWH9_CreateGoodsReceiptPOLinesScreen_TextWarehouse_validate_after

```
IF (SELECT ISNULL([@SCANEMPWH].U_ToWh,'')
FROM [@SCANEMPWH]
WHERE U_empID = $[Employee.EmployeeID]) <> $[TextWarehouse]
UPDATE [@SCANEMPWH]
SET [U_ToWh]=$[TextWarehouse]
WHERE U_empID = $[Employee.EmployeeID]
```

Check if item has been ordered/open quantity from this supplier
Query name: BXMobileWH9_CreateGoodsReceiptPOLinesScreen_TextItem_validate

```
DECLARE @itemcode nvarchar(MAX)
SET @itemcode = $[TextItem]
IF NOT EXISTS (SELECT T0.ItemCode FROM OITM T0 WHERE T0.ItemCode=@itemcode)
SELECT @itemcode=T1.ItemCode FROM OBCD T1 WHERE T1.BcdCode=@itemcode
IF @itemcode<>'' AND (SELECT ISNULL(SUM(T1.[OpenQty]),0) FROM OPOR T0
INNER JOIN POR1 T1 ON T0.[DocEntry] = T1.[DocEntry] WHERE T1.[ItemCode]
=@itemcode AND T0.[CardCode] =
LEFT($[TextSupplier],PATINDEX('%*%',$[TextSupplier])-1))=0
SELECT 'There is no Purchase Order: '+ @itemcode AS 'Message$', 'E' AS
'MessageType$'
ELSE SELECT '' AS dummy
```

Check if item has been ordered/open quantity from this supplier

Query name: BXMobileWH9_CreateGoodsReceiptPOLinesScreen_TextItem_validate

```
DECLARE @itemcode nvarchar(MAX)
SET @itemcode = $[TextItem]
IF NOT EXISTS (SELECT T0.ItemCode FROM OITM T0 WHERE T0.ItemCode=@itemcode)
SELECT @itemcode=T1.ItemCode FROM OBCD T1 WHERE T1.BcdCode=@itemcode
IF @itemcode<>'' AND (SELECT ISNULL(SUM(T1.[OpenQty]),0) FROM OPOR T0
INNER JOIN POR1 T1 ON T0.[DocEntry] = T1.[DocEntry] WHERE T1.[ItemCode]
=@itemcode AND T0.[CardCode]
=LEFT($[TextSupplier],PATINDEX('%*%',$[TextSupplier])-1))=0
SELECT 'There is no Purchase Order: '+ @itemcode AS 'Message$', 'E' AS
'MessageType$'
ELSE SELECT '' AS dummy
```

After item code has been entered write ' ' to quantity and press Add
Query name: BXMobileWH9_CreateGoodsReceiptPOLinesScreen_TextItem_validate_after

```
IF $[TextItem]<>'' SELECT '' AS TextQuantity, 'ButtonAdd' AS Click$
```

CreateGoodsReceiptPOQuantitiesSerialScreen

Add the following record to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	CreateGoodsReceiptPOQuantitiesSerialScreen

Create a new user table ‘SCANEMPBL’ with the following fields: • Employee ID (empID) • To Warehouse (ToBL)

Load bin location from employee, fill EAN bar code
Query name: BXMobileWH9_CreateGoodsReceiptPOQuantitiesSerialScreen_Load

```
SELECT
(SELECT [@SCANEMPBL].U_ToBL FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextBinLocation',
(SELECT CodeBars FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode'
```

Save bin location to employee
Query name:
BXMobileWH9_CreateGoodsReceiptPOQuantitiesSerialScreen_TextBinLocation_validate_after

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

After serial entry press OK to go back

Query name: BXMobileWH9_CreateGoodsReceiptPOQuantitiesSerialScreen_TextSerial_validate_after

```
IF $[TextSerial]<>'' SELECT 'OK' AS Click$
```

CreateGoodsReceiptPOQuantitiesNormalScreen

Add the following record to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	GoodsReceiptPOQuantitiesNormalScreen
FieldName	**Visible**	**Screen**	
TextUoM	NO	GoodsReceiptPOQuantitiesNormalScreen	

Create a new user table ‘SCANEMPBL’ with the following fields:

- Employee ID (empID)
- To Warehouse (ToBL)

Load last location for employee, EAN bar code, set quantity to 1 and focus on quantity
Query name: BXMobileWH9_CreateGoodsReceiptPOQuantitiesNormalScreen_Load

```
SELECT
(SELECT CodeBars FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode',
(SELECT [@SCANEMPBL].U_ToBL FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextBinLocation', '1' AS
'TextQuantity' ,'TextQuantity' AS Click$
```

Save bin location to employee
Query name:
BXMobileWH9_CreateGoodsReceiptPOQuantitiesNormalScreen_TextBinLocation_validate_after

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

After add button press OK to go back
Query name: BXMobileWH9_CreateGoodsReceiptPOQuantitiesNormalScreen_ButtonAdd_click_after

```
SELECT 'OK' AS Click$
```

Transfer Stocks

TransferStocksScreen

Add the following record to the Customization Fields user table:

FieldName	Visible	Screen
TextBatch	NO	CreateGoodsReceiptPOQuantitiesSerialScreen

Modify the list contents: show EAN bar code instead of item code
Query name: BXMobileWH9_TransferStocksScreen_DataRepeater_InternalDataLoad

```
SELECT
SUBSTRING(S.splitdata,1,15)+'EAN:'+ ISNULL(C.CodeBars,'')
AS [DataRepeater.UIItemCode]
FROM dbo.SplitStringForDataRepeater($[DataRepeater.UIItemCode]) S, OITM C
WHERE C.ItemCode=SUBSTRING(S.splitdata,1,PATINDEX('%*%',S.splitdata)-2)
```

Focus on item field after returning to this screen
Query name: BXMobileWH9_TransferStocksScreen_Activate

```
SELECT 'TextItem' AS Click$
```

TransferStocksQuantitiesSerialScreen

Add the following record to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	TransferStocksQuantitiesSerialScreen

Create a new user table 'SCANEMPBL' with the following fields:

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

Fill to location from saved, load bar code into EAN code
Query name: BXMobileWH9_TransferStocksQuantitiesSerialScreen_Load

```
SELECT
(SELECT [@SCANEMPBL].U_ToBL
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextToLocation',
(SELECT CodeBars FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode'
```

Save to location for employee
Query name: BXMobileWH9_TransferStocksQuantitiesSerialScreen_TextToLocation_validate_after

```
IF
(SELECT ISNULL([@SCANEMPBL].U_ToBL,'')
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) <> $[TextToLocation]
UPDATE [@SCANEMPBL]
```

```
SET [U_ToBL]=$[TextToLocation] WHERE U_empID = $[Employee.EmployeeID]
```

After serial number entered, press OK to go back
Query name: BXMobileWH9_TransferStocksQuantitiesSerialScreen_TextSerialNumber_validate_after

```
IF $[TextSerialNumber]<>'' SELECT 'OK' AS Click$
```

TransferStocksQuantitiesNormalScreen

Add the following record to the Customization Fields user table:

FieldName	**Label**	**ReadOnly**	**Screen**
EanCode	EAN	YES	TransferStocksQuantitiesNormalScreen
FieldName	**Visible**	**Screen**	
TextUoM	NO	TransferStocksQuantitiesNormalScreen	

Create a new user table 'SCANEMPBL' with the following fields:

- Employee ID (empID)
- To Warehouse (ToBL)

Load EAN bar code and to location, set quantity to 1 and focus on quantity
Query name: BXMobileWH9_TransferStocksQuantitiesNormalScreen_Load

```
SELECT
(SELECT CodeBars
FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode',
(SELECT [@SCANEMPBL].U_ToBL
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextToLocation','1' AS
'TextQuantity','TextQuantity' AS Click$
```

Save to location for employee
Query name: BXMobileWH9_TransferStocksQuantitiesNormalScreen_TextToLocation_validate_after

```
IF (SELECT ISNULL([@SCANEMPBL].U_ToBL,'')
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) <> $[TextToLocation]
UPDATE [@SCANEMPBL]
SET [U_ToBL]=$[TextToLocation]
WHERE U_empID = $[Employee.EmployeeID]
```

Press OK to go back after Add button
Query name: BXMobileWH9_TransferStocksQuantitiesNormalScreen_ButtonAdd_click_after

```
SELECT 'OK' AS Click$
```

Mass Transfer

TransferStocksMassScreen

Create a new user table 'SCANEMPBL' with the following fields:

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

Load saved bin location from and to
Query name: BXMobileWH9_TransferStocksMassScreen_Load

```
SELECT
(SELECT [@SCANEMPBL].U_FromBL FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextWHBinLocationFrom',
(SELECT [@SCANEMPBL].U_ToBL
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextWHBinLocationTo'
```

Save from bin location
Query name: BXMobileWH9_TransferStocksMassScreen_TextWHBinLocationFrom_validate_after

```
IF $[TextWHBinLocationFrom] <>''
AND (SELECT ISNULL([@SCANEMPBL].U_FromBL,'')
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) <> $[TextWHBinLocationFrom]
UPDATE [@SCANEMPBL]
SET [U_FromBL]=$[TextWHBinLocationFrom]
WHERE U_empID = $[Employee.EmployeeID]
```

Save to bin location
Query name: BXMobileWH9_TransferStocksMassScreen_TextWHBinLocationFrom_validate_after

```
IF $[TextWHBinLocationTo] <>''
AND (SELECT ISNULL([@SCANEMPBL].U_ToBL,'')
FROM [@SCANEMPBL]
WHERE U_empID =$[Employee.EmployeeID]) <> $[TextWHBinLocationTo]
UPDATE [@SCANEMPBL]
SET [U_ToBL]=$[TextWHBinLocationTo]
WHERE U_empID = $[Employee.EmployeeID]
```

Modify list contents
• [DataRepeater. UIItemCode] Item code + name • [DataRepeater.UIOnHandQuantity] Bar code (EAN) + quantity Query name: BXMobileWH9_TransferStocksMassScreen_DataRepeater_InternalDataLoad

```
-- Modify list contents
-- [DataRepeater. UIItemCode] Item code + name
-- [DataRepeater.UIOnHandQuantity] Bar code (EAN) + quantity
```

```
SELECT C.CodeBars + ' ' +SUBSTRING(S.splitdata2,1,50) AS
[DataRepeater.UIOnHandQuantity]
FROM dbo.SplitStringForDataRepeater2($[DataRepeater.UIItemCode], $
[DataRepeater.UIOnHandQuantity]) S, OITM C
WHERE C.ItemCode=SUBSTRING(S.splitdata,1,PATINDEX('%*%',S.splitdata)-1)
```

Transfer Stock Request

TransferStockRequestLineScreen

Add the following record to the Customization Fields user table:

FieldName	Visible	Screen
TextBatch	NO	TransferStockRequestLineScreen

Modify list contents, add EAN bar code
Query name: BXMobileWH9_TransferStockRequestLineScreen_DataRepeater_InternalDataLoad

```
SELECT SUBSTRING(S.splitdata,7,15)+' EAN:'+ ISNULL(C.CodeBars,'') AS
[DataRepeater.UIItem]
FROM dbo.SplitStringForDataRepeater($[DataRepeater.UIItem]) S, OITM C
WHERE C.ItemCode=SUBSTRING(S.splitdata,7,PATINDEX('%*%',S.splitdata)-8)
```

After Item Code has been entered, set quantity to ' ' and press Add
Query name: BXMobileWH9_TransferStockRequestLineScreen_TextItem_validate_after

```
IF $[TextItem]<>'' SELECT '' AS TextQuantity, 'ButtonAdd' AS Click$
```

Item code scanning is mandatory (before pressing Add button)
Query name: BXMobileWH9_TransferStockRequestLineScreen_ButtonAdd_click

```
IF $[TextItem] =''
SELECT 'You must scan an item!' AS 'Message$', 'E' AS 'MessageType$',
'TextItem'
AS Click$
```

Item code scanning is mandatory (before pressing Update button)
Query name: BXMobileWH9_TransferStockRequestLineScreen_ButtonUpdate_click

```
IF $[TextItem] =''
SELECT 'You must scan an item!' AS 'Message$', 'E' AS 'MessageType$',
'TextItem'
AS Click$
```

Focus on Item field (when returning to this screen)
Query name: BXMobileWH9_TransferStockRequestLineScreen_Activate

```
SELECT 'TextItem' AS Click$
```

4.1.5.2. TransferStockRequestsQuantitiesSerialScreen

Add the following record to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	TransferStockRequestsQuantitiesSerialScreen

Create a new user table ‘SCANEMPBL’ with the following fields:

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

Set to location from saved, set EAN code from barcode
Query name: BXMobileWH9_TransferStockRequestsQuantitiesSerialScreen_Load

```
SELECT
(SELECT [@SCANEMPBL].U_ToBL FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextToLocation',
(SELECT CodeBars FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode'
```

Save to location if changed
Query name:
BXMobileWH9_TransferStockRequestsQuantitiesSerialScreen_TextToLocation_validate_after

```
IF (SELECT ISNULL([@SCANEMPBL].U_ToBL,'')
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID])<> $[TextToLocation]
UPDATE [@SCANEMPBL]
SET [U_ToBL]=$[TextToLocation] WHERE U_empID = $[Employee.EmployeeID]
```

After serial number has been entered press OK to go back
Query name:
BXMobileWH9_TransferStockRequestsQuantitiesSerialScreen_TextSerialNumber_validate_after

```
IF $[TextSerialNumber]<>'' SELECT 'OK' AS Click$
```

TransferStockRequestsQuantitiesNormalScreen

Add Customization Fields user table records:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	TransferStockRequestsQuantitiesNormalScreen
FieldName	**Visible**	**Screen**	
TextUoM	NO	TransferStockRequestsQuantitiesNormalScreen	

Create a new user table ‘SCANEMPBL’ with the following fields:

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

Fill EanCode, To location, set 1 to quantity and focus on quantity
Query name: BXMobileWH9_TransferStockRequestsQuantitiesNormalScreen_Load

```
SELECT
(SELECT CodeBars FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode',
(SELECT [@SCANEMPBL].U_ToBL
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) AS 'TextToLocation', '1' AS
'TextQuantity' ,'TextQuantity' AS Click$
```

Save to bin location for next time
Query name:
BXMobileWH9_TransferStockRequestsQuantitiesNormalScreen_TextToLocation_validate_after

```
IF (SELECT ISNULL[@SCANEMPBL] (.U_ToBL,'')
FROM [@SCANEMPBL]
WHERE U_empID = $[Employee.EmployeeID]) <> $[TextToLocation]
UPDATE [@SCANEMPBL]
SET [U_ToBL]=$[TextToLocation]
WHERE U_empID = $[Employee.EmployeeID]
```

After Add press OK to go back
Query name: BXMobileWH9_TransferStockRequestsQuantitiesNormalScreen_ButtonAdd_click_after

```
SELECT 'OK' AS Click$
```

Picking Screen

Picking Screen

Add the following record to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
BO_U_Tour ID	TourID	YES	PickingLinesScreen

Show TourID in the list in the place of date field
Query name: BXMobileWH9_PickingScreen_DataRepeater_InternalDataLoad

```
SELECT SUBSTRING(S.splitdata2,1,50)+ ' Tour: ' + C.U_TourID AS
[DataRepeater.UIPickDate]
FROM dbo.SplitStringForDataRepeater2($[DataRepeater.UIPickListNo],
$[DataRepeater.UIPickDate]) S, OPKL C
```

```
WHERE C.AbsEntry= CAST(SUBSTRING(S.splitdata, 11,5) AS INT)
```

Picking Lines Screen

Add the following record to the Customization Fields user table. The Tour ID field will be shown on the screen and the Bin Code field will be editable.

FieldName	Label	ReadOnly	Screen
BO_U_ TourID	TourID	YES	PickingLinesScreen
TextBinCode	Item	NO	PickingLinesScreen

Fill TourID field on top of the screen
Query name: BXMobileWH9_PickingLinesScreen_Load

```
SELECT
(SELECT OPKL.U_TourID FROM OPKL WHERE OPKL.AbsEntry =
CAST(LEFT($[TextPickList],PATINDEX('%*%',$[TextPickList])-1) AS INT)) AS
[BO_U_TourID]
```

Mandatory to scan/enter item in bin code field
Query name: BXMobileWH9_PickingLinesScreen_OK_clicked

```
IF $[TextBinCode] =''
SELECT 'Scan an item!' AS 'Message$', 'E' AS 'MessageType$', 'TextBinCode'
AS Click$
```

Show EAN bar code in list
Query name: BXMobileWH9_PickingLinesScreen_DataRepeater_InternalDataLoad

```
SELECT ' EAN: ' + C.CodeBars AS [DataRepeater.UIBinCode]
FROM dbo.SplitStringForDataRepeater2($[DataRepeater.UIItem],
$[DataRepeater.UIBaseType]) S, OITM C
WHERE C.ItemCode=SUBSTRING(S.splitdata,1,PATINDEX('%*%',S.splitdata)-1)
```

Focus on bin code field
Query name: BXMobileWH9_PickingLinesScreen_Activate

```
SELECT 'TextBinCode' AS Click$
```

PickingLinesPickSerialScreen

Add new EanCode field to screen, set bin field to read only. Add the following records to the Customization Fields user table:

FieldName	Label	ReadOnly	Screen
EanCode	EAN	YES	PickingLinesPickSerialScreenw
FieldName	**Label**	**Screen**	

FieldName	**Label**	**ReadOnly**	**Screen**
TextBinLocation	YES	PickingLinesPickSerialScreen	

Show the bin location and the EAN code of the item on the screen
Query name: BXMobileWH9_PickingLinesPickSerialScreen_Load

```
SELECT $[TextRecBin] AS 'TextBinLocation',
(SELECT CodeBars FROM OITM
WHERE ItemCode=SUBSTRING($[TextItem],1,PATINDEX('%*%',$[TextItem])-2) ) AS
'EanCode'
```

Press the Post button after the serial number is scanned if the open quantity is zero
Query name: BXMobileWH9_PickingLinesPickSerialScreen_TextSerialNumber_validate_after

```
IF $[TextSerialNumber]<>''
IF CAST(LEFT($[TextOpenQty],PATINDEX('% %',$[TextOpenQty]+' ')-1) AS INT)=0
SELECT 'OK' AS 'Click$'
ELSE SELECT 'OK##Option' AS 'Click$'
```

PickingLinesPickNormalScreen

Hide UoM field from screen. Add the following record to the Customization Fields user table:

FieldName	**Label**	**ReadOnly**	**Screen**
EanCode	EAN	YES	PickingLinesPickNormalScreen
FieldName	**Visible**	**Screen**	
TextUoM	NO	PickingLinesPickNormalScreen	
FieldName	**ReadOnly**	**Screen**	
TextBinLocation	YES	PickingLinesPickNormalScreen	

Show the bin location and the EAN code of the item on the screen
Query name: BXMobileWH9_PickingLinesPickNormalScreen_Load

```
SELECT $[TextRecBin] AS 'TextBinLocation',
(SELECT CodeBars FROM OITM
WHERE ItemCode=$[SelectedPickListLine.ItemCode]) AS 'EanCode',1 AS
'TextQuantity', 'TextQuantity' AS Click$
```

After quantity has been entered, press OK to book and then back button
Query name: BXMobileWH9_PickingLinesPickNormalScreen_TextQuantity_validate_after

```
IF CAST(LEFT($[TextOpenQty],PATINDEX('% %',$[TextOpenQty]+' ')-1) AS INT)-
CAST($[TextQuantity] AS INT)<=0
SELECT 'OK' AS 'Click$'
ELSE SELECT 'OK##Option' AS 'Click$'
```

Goods Receipt

GoodsReceiptLinesScreen

Hide UoM field from screen. Add the following record to the Customization Fields user table:

FieldName	Visible	Screen
TextUoM	NO	PickingLinesScreen

Sets price, currency and account no fields
Query name: bx_mobile_wh9_goodsreceipt_pricing

```
SELECT
    ISNULL(T1.Price,0) AS 'BXITPRC',
    ISNULL(T1.Currency,'FT') AS 'BXITCURR',
    '5911' AS 'BXITACCN'
FROM OITM T0
JOIN ITM1 T1 ON T0.ItemCode=T1.ItemCode AND T1.Pricelist='1'
WHERE T0.[ItemCode] =[%3]
```

Create a new user table ‘SCANEMPWH’ with the following fields:

- Employee ID (empID)
- To Warehouse (ToWh)

Load warehouse from employee's saved value, focus on item
Query Name: BXMobileWH9_GoodsReceiptLinesScreen_Load

```
SELECT
(SELECT [@SCANEMPWH].U_ToWh FROM [@SCANEMPWH]
WHERE U_empID =$[Employee.EmployeeID]) AS 'TextWarehouse', 'TextItem' AS
Click$
```

Save warehouse code for next time
Query Name: Save warehouse code for next time

```
IF (SELECT ISNULL([@SCANEMPWH].U_ToWh,'') FROM [@SCANEMPWH]
WHERE U_empID = $[Employee.EmployeeID]) <> $[TextWarehouse]
UPDATE [@SCANEMPWH] SET [U_ToWh]=$[TextWarehouse] WHERE U_empID =
$[Employee.EmployeeID]
```

After item code has been entered, automatically set quantity to ' ' and press add
Query Name: BXMobileWH9_GoodsReceiptLinesScreen_TextItem_validate_after

```
IF $[TextItem]<>'' SELECT '' AS TextQuantity, 'ButtonAdd' AS Click$
```

Change the contents of the list: [DataRepeater.UIItem] Item code + name, [DataRepeater.UIWarehouse] Bar code (EAN)
Query Name: BXMobileWH9_GoodsReceiptLinesScreen_DataRepeater_InternalDataLoad

```
SELECT
    SUBSTRING(S.splitdata,7,50) AS [DataRepeater.UIItem],
    C.CodeBars AS [DataRepeater.UIWarehouse]
FROM dbo.SplitStringForDataRepeater($[DataRepeater.UIItem]) S, OITM C
WHERE C.ItemCode=SUBSTRING(S.splitdata,7,PATINDEX('%*%',S.splitdata)-8)
```

Focus on item code (on return to this screen)
Query Name: BXMobileWH9_GoodsReceiptLinesScreen_Activate

```
SELECT 'TextItem' AS Click$
```

Query Stocks

QueryStocksScreen

Hide batch field from screen. Add the following record to the Customization Fields user table:

FieldName	Visible	Screen
TextBatch	NO	QueryStocksScreen

Create a new user table 'SCANEMPWH' with the following fields:

- Employee ID (empID)
- To Warehouse (ToWh)

Load default warehouse for employee, focus on Item field
Query Name: BXMobileWH9_QueryStocksScreen_Load

```
SELECT
(SELECT [@SCANEMPWH].U_ToWh FROM [@SCANEMPWH] WHERE U_empID =
$[Employee.EmployeeID]) AS 'TextWHBinLocation', 'TextItem' AS Click$
```

Modify list contents (show bar code)
Query Name: BXMobileWH9_QueryStocksScreen_DataRepeater_InternalDataLoad

```
SELECT SUBSTRING(S.splitdata,1,50) +' VK: '+ C.CodeBars AS
[DataRepeater.UIItemCode]
FROM dbo.SplitStringForDataRepeater($[DataRepeater.UIItemCode]) S, OITM C
WHERE C.ItemCode=SUBSTRING(S.splitdata,1,PATINDEX('%*%',S.splitdata)-1)
```

Appendix

Helper SQL procedures

```
-- This function is a helper for Produmex Scan list splitting
CREATE FUNCTION [dbo].[SplitStringForDataRepeater]
```

```
( @string NVARCHAR(MAX) )
RETURNS @output TABLE(splitdata NVARCHAR(MAX) )
BEGIN
DECLARE @START INT, @END INT
SELECT @START = 1, @END = CHARINDEX('##', @string)
WHILE @START < LEN(@string) + 1 BEGIN
IF @END = 0
SET @END = LEN(@string) + 2
INSERT INTO @output (splitdata)
VALUES(SUBSTRING(@string, @START, @END - @START))
SET @START = @END + 2
SET @END = CHARINDEX('##', @string, @START)
END
RETURN
END
```

```
-- This function is a helper for Produmex Scan list splitting
CREATE FUNCTION [dbo].[SplitStringForDataRepeater2]
( @string NVARCHAR(MAX), @string2 NVARCHAR(MAX) )
RETURNS @output TABLE(splitdata NVARCHAR(MAX) , splitdata2 NVARCHAR(MAX))
BEGIN
DECLARE @START INT, @END INT , @start2 INT, @end2 INT
SELECT @START = 1, @END = CHARINDEX('##', @string) , @start2 = 1, @end2 =
CHARINDEX('##', @string2)
WHILE @START < LEN(@string) + 1 BEGIN
IF @END = 0
BEGIN
SET @END = LEN(@string) + 2
SET @end2 = LEN(@string2) + 2
END
INSERT INTO @output (splitdata, splitdata2)
VALUES(SUBSTRING(@string, @START, @END - @START),SUBSTRING(@string2,
@start2, @end2 - @start2))
SET @START = @END + 2
SET @END = CHARINDEX('##', @string, @START)
SET @start2 = @end2 + 2
SET @end2 = CHARINDEX('##', @string2, @start2)
END
RETURN
END
```