Manipulating DataRepeater (List)

There are list boxes on the screens in which items, SAP documents, serial or batch numbers appear. In the Produmex Scan these list objects are called: DateRepeater.

In one row you can usually find multiple data. For example Itemcode, Location, Document Line Number.

The pieces of information for one type of data are stored concatenated in a single string, separated by: “##”.

There are two parameters that we have for one type of data:

- You can find the concatenated value in the parameter that starts with: DataRepeater.UI
- You can find the value from the selected row in the parameter that starts with: DataRepeater.SelectedUI

Example:

```
$[DataRepeater.UIWarehouse] = WH: 01##WH: 02##WH: 03
$[DataRepeater.SelectedUIWarehouse] = WH: 01
```

If you would like to manipulate the appearing content then you should convert the concatenated string into an SQL table with our example splitter functions shown below. This table can be used in a query in order to load the data you need.

As a final step define the column in your query. The column name should be the same as the name of the parameter that you want to replace, for example:

```
SELECT
    'NewInfo' as "DataRepeater.UIRemarks"
FROM
.....
```

You will need a splitter function, to separate data. If it does not exist, then you can create them with these queries:

SQL

```
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
```

```
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
```

HANA

```
CREATE FUNCTION SPLITSTRINGFORDATAREPEATER2 (strstring nvarchar(5000),
strstring2 nvarchar(5000) )
RETURNS TABLE (splitdata nvarchar(5000) , splitdata2 nvarchar(5000) )
LANGUAGE SQLSCRIPT AS
BEGIN
    DECLARE strstart INT;
    DECLARE strend INT;
    DECLARE strstart2 INT;
    DECLARE strend2 INT;
    declare _items nvarchar(5000) ARRAY;
    declare _index integer;
    _index := 1;
    strstart := 1;
    strend := LOCATE(strstring, '##');
    strstart2 := 1;
    strend2 := LOCATE(strstring2, '##');
    WHILE strstart < LENGTH(strstring) + 1 DO
        IF strend = 0 THEN
            BEGIN
                strend := LENGTH(strstring) + 2;
                strend2 := LENGTH(strstring2) + 2;
            END;
        END IF;
        _items[:_index] := SUBSTRING(strstring, strstart, strend - strstart)
|| '|||' || SUBSTRING(strstring2, strstart2, strend2 - strstart2);
        _index := :_index + 1;
        strstart := strend + 2;
        strend := LOCATE(strstring, '##', strstart);
        strstart2 := strend2 + 2;
        strend2 := LOCATE(strstring2, '##', strstart2);
    end while;

    rst = UNNEST(:_items) AS ("items");
    RETURN
        SELECT SUBSTRING("items", 1, LOCATE("items", '|||') - 1) as
"SPLITDATA", SUBSTRING("items", LOCATE("items", '|||') + 3, (LENGTH("items")
- LOCATE("items", '|||') ) ) as "SPLITDATA2" FROM :rst;
 END;
```

Add a string

This customization will add a string to the Warehouse field on Sales Order Issue Lines screen.

Query name: *BXMobileWH9_SalesIssueLinesScreen_DataRepeater_InternalDataLoad*

SQL

```
SELECT
    S.splitdata + ' Additional Info' AS [DataRepeater.UIWarehouse] from
    dbo.SplitStringForDataRepeater($[DataRepeater.UIWarehouse]) S
```

HANA

```
SELECT
    "S"."SPLITDATA" || ' Additional Info ' || AS "DataRepeater.UIWarehouse"
FROM
    SPLITSTRINGFORDATAREPEATER2(
       $[DataRepeater.UIWarehouse], '') AS "S"
```

Add information from RDR1

This customization will add information from RDR1 table by VisOrder.

Note: On some screens the sequence of VisOrder may start with 1 instead of 0 (in contrast with the SAP database table). In this case you can either reduce the splitter value with 1 or increase the VisOrder SAP field with 1 in the query.

We can load DocNumber from $[TextSalesOrder] field, and we can get the exact number with SUBSTRING. SplitStringForDataRepeater2 function splits UIWarehouse UILineNumber from the datarepeater, and then we can use them to connect to SAP table, and we can use the splitted values in the query.

Query name: *BXMobileWH9_SalesIssueLinesScreen_DataRepeater_InternalDataLoad*

SQL

```
SELECT
    S.splitdata + ' - ' + ItemCode  AS [DataRepeater.UIWarehouse]      FROM
    dbo.SplitStringForDataRepeater2(        $[DataRepeater.UIWarehouse] ,
        $[DataRepeater.UILineNumber] ) S
    LEFT JOIN
    (SELECT
            '#'+ cast(VisOrder + 1 as varchar(10)) as ln,
            ItemCode
        FROM
            RDR1 left join ORDR on RDR1.DocEntry = ORDR.DocEntry
        WHERE
            ORDR.DocNum = SUBSTRING($[TextSalesOrder] ,0,CHARINDEX('*',
$[TextSalesOrder] )) )              as RDR1Line
    ON S.splitdata2 = RDR1Line.ln
```

HANA

```
SELECT
    "S"."SPLITDATA" || ' - ' || "ItemCode" AS "DataRepeater.UIWarehouse"
FROM
    SPLITSTRINGFORDATAREPEATER2(
       $[DataRepeater.UIWarehouse] ,
           $[DataRepeater.UILineNumber] ) AS "S"
    LEFT JOIN
    (SELECT
            '#' || cast("VisOrder" + 1 as varchar(10)) as "ln",
             "ItemCode"
        FROM
            "RDR1" left join "ORDR" on "RDR1"."DocEntry" = "ORDR"."DocEntry"
        WHERE
            "ORDR"."DocNum" = CAST( SUBSTRING($[TextSalesOrder] ,0,
LOCATE($[TextSalesOrder], '*' ) - 2) AS int)) as "RDR1Line"
    ON "S"."SPLITDATA2" = "RDR1Line"."ln"
```

Adding extra lines to screens

This customization will add extra lines to the necessary screen beside the default lines.

Example
In this example we will add 3 extra lines to the *Goods Receipts PO* screen by running the necessary SAP user query. The extra lines are:

1. Posting Date
2. Document Date
3. Ship to

1. We list the 3 extra lines in the Customization Fields window and fill in the necessary columns. In the *Position Data* column we also indicate that we wish to add 3 extra lines.

2. We run the necessary query.

Query name: *BXMobileWH9_GoodsReceiptPOScreen_DataRepeater_InternalDataLoad*

SQL

```
SELECT
    REPLACE( S.splitdata , 'PO #', ''),
    'Posting Date ' + FORMAT( OPOR.DocDate, 'dd/MM/yyyy', 'en-US' ) as
[DataRepeater.UICust01],
    'Document Date ' + FORMAT( OPOR.TaxDate, 'dd/MM/yyyy', 'en-US' ) as
[DataRepeater.UICust02],
    'Ship to ' + OPOR.Address2 as [DataRepeater.UICust03]

from
     dbo.SplitStringForDataRepeater( $[DataRepeater.UIDocNum]  ) as S
     LEFT JOIN OPOR on REPLACE( S.splitdata , 'PO #', '') = OPOR.DocNum
```

HANA

```
SELECT
    REPLACE( "S"."SPLITDATA" , 'PO #', ''),
    'Posting Date ' || to_char( "OPOR"."DocDate", 'DD/MM/YYYY') as
"DataRepeater.UICust01",
    'Document Date ' || to_char( "OPOR"."TaxDate", 'DD/MM/YYYY') as
"DataRepeater.UICust02",
    'Ship to ' || "OPOR"."Address2" as "DataRepeater.UICust03"

from
     SPLITSTRINGFORDATAREPEATER2( $[DataRepeater.UIDocNum], '' ) as "S"
     LEFT JOIN "OPOR" on REPLACE( "S"."SPLITDATA" , 'PO #', '') =
"OPOR"."DocNum";
```

SAP user query:

3. After executing the query, the 3 extra lines will appear on the *Goods Receipt PO* screen as follows: