

---

# ENTERPRISE RESOURCE PLANNING SYSTEMS

---

**Documentation:** Assignment 2

**Authors:** Martin Zaefferer, Thomas Karanatsios

**Matrikel Numbers:** 11053585, 11032154

## CONTENT

---

1. Introduction .....	1
2. Matrikelnumber Check .....	2
2.1. Create a Codeunit (Name: MatNumCodeUnit) .....	2
2. 2. Create a blank page.....	2
3. Managing Employee Data .....	3
3.1. Computing Month and Year from LabourDate.....	3
3.2. Making TotalHours a FlowField of type sum .....	4
3.3. Creating a Page with a embedded List part.....	5
4. Codeunits.....	6
4.1.Codeunit that counts records .....	6
4.2. Codeunit that Deletes and Creates Records.....	7
5. Writing to a File with XML Port.....	8

## 1. INTRODUCTION

---

In this document we describe the solutions for the 4 different tasks of exercise 2 in the Enterprise Resource Planning lecture. In the first task we have to create an unbounded page where we could enter our "Matrikelnumber". We also must check if the "Matrikelnumber" is valid with an Action trigger. The second task is to create different tables and create a relation between them with the Primary Keys given in the Assignment. The third task is to create a Codeunit which counts the number of records inside a Table and the fourth task was to create an xml port which you can use to import and export text files.

## 2. MATRIKELNUMBER CHECK

---

In this task as explained in the introduction an unbounded page is to be created, which checks "Matrikelnumbers" of users and proofs if the input is valid. The following steps are necessary to create such an unbounded page.

### 2.1. CREATE A CODEUNIT (NAME: MATNUMCODEUNIT)

---

- 2.1.1. Create a function for the Codeunit, in the "Globals" window
- 2.1.2. The function should have two locals: The argument passed to it (Text variable) and a return value (boolean) which tells if the number input is valid.
- 2.1.3. Also three further variables will be created as globals for the Codeunit: I, TestNum, and buff. All three of them are integers, the TestNum is an integer array. To change a variable to an array, simply open the properties window of the variable, and enter a dimension (here: 10)
- 2.1.4. Enter the necessary code to check the text input for its validity

### 2. 2. CREATE A BLANK PAGE

---

- 2.2.1. Enter "Global" variables for this page: MatNum (DataType: Text), Res (DataType: Boolean), MatCode (DataType: Codeunit, Subtype: MatNumCodeUnit)
- 2.2.2. Enter a container into the page
- 2.2.3. Enter a field into the page, as part of the container
- 2.2.4. Choose "MatNum" variable as source expression for the new field
- 2.2.5. Right click an empty line in the page designer
- 2.2.6. Go to "Actions"
- 2.2.7. Create an action container and a corresponding action in the action designer
- 2.2.8. Go to the "Code" view of the new action, and enter the code to call the function created in step 1 by using the MatCode variable and the function name.
- 2.2.9. Use the resulting boolean to give a message to the user.

### 3. MANAGING EMPLOYEE DATA

For creating the 3 tables you can use the table designer which simplifies the work a lot. The more difficult thing on this task was to define the primary key and here is the description to do this:

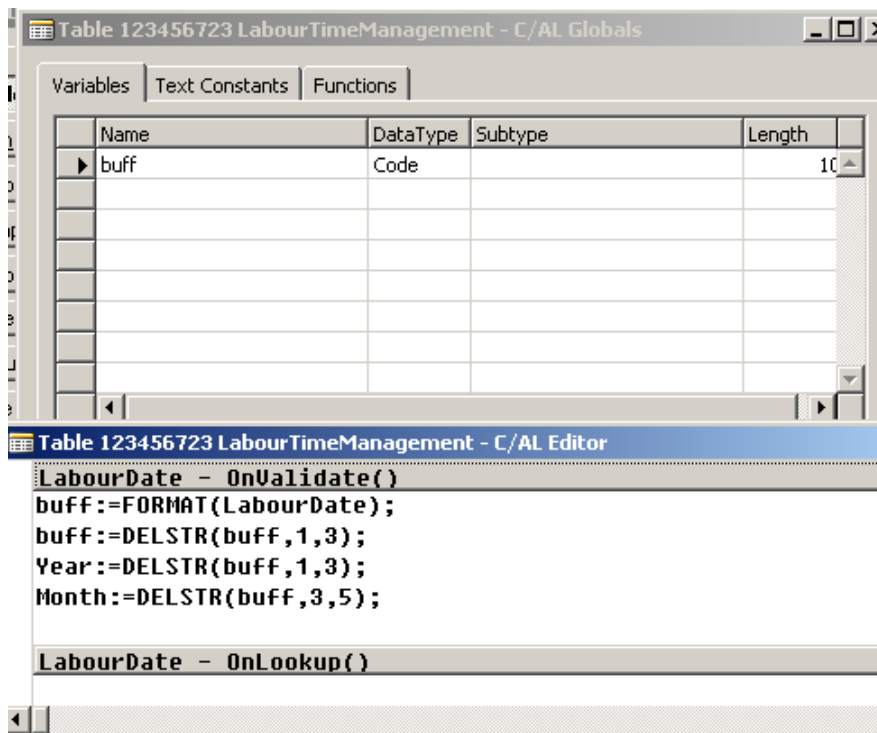
#### To define a primary key

- On the Tools menu, click **Object Designer**.
- Click **Table**.
- Select a table and then click **Design**.
- Select the first empty line and then from the **View** menu, click **Keys**.
- In the first line of the **Keys window**, enter the primary key as a comma-delimited list; for example, "ID Number, Name".

#### To create a table relationship

- Open Object Designer and click **Table** to see a list of the tables.
- Select a table and then click **Design**.
- The table opens in **Table Designer**.
- Select the field for which you want to create a relationship and view the **Properties** window.
- Set the **TableRelation** property value by using the following steps:
  - In the **TableRelation** property value, click the **AssistEdit** button.
  - The **Table Relation** window opens.
  - Click the **AssistEdit** in the **Table** cell, select a table, and then click **OK**.
  - Click the **AssistEdit** in the **Field** cell, select a field, and then click **OK**.
  - Set the **Condition** or **Filter** fields, as needed.
  - Click **OK**.

#### 3.1. COMPUTING MONTH AND YEAR FROM LABOURDATE



The screenshot shows two windows from the SAP ABAP Editor. The top window, titled 'Table 123456723 LabourTimeManagement - C/AL Globals', displays a table with the following data:

Name	Data Type	Subtype	Length
buff	Code		10

The bottom window, titled 'Table 123456723 LabourTimeManagement - C/AL Editor', shows the following source code:

```

LabourDate - OnValidate()
buff:=FORMAT(LabourDate);
buff:=DELSTR(buff,1,3);
Year:=DELSTR(buff,1,3);
Month:=DELSTR(buff,3,5);

LabourDate - OnLookup()
  
```

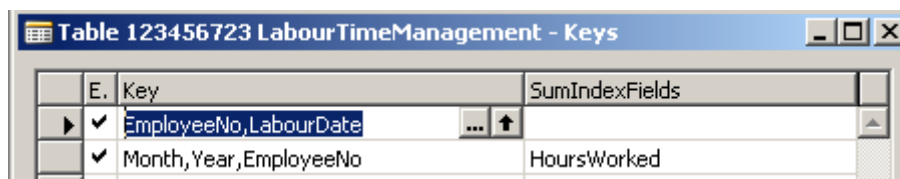
Picture 1: Source Code Month and Year

### 3.2. MAKING TOTALHOURS A FLOWFIELD OF TYPE SUM

#### To create a FlowField

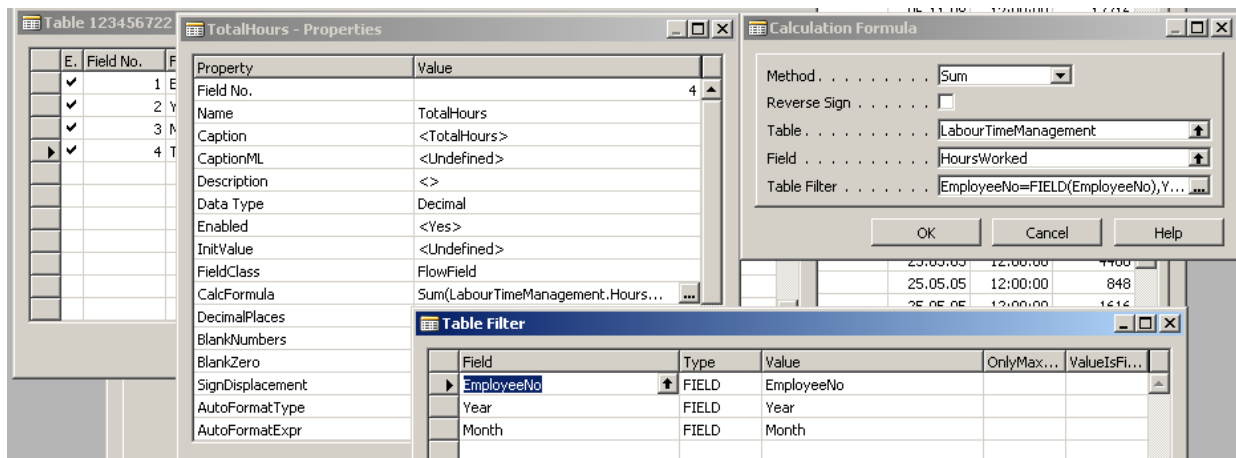
- On the **Tools** menu, click **Object Designer**. Open the table that you want to add a FlowField to in the Table Designer.
- Select the line for the field that you want to define as a FlowField, such as the **Amount** field.
- On the **View** menu, click **Properties**. The **Properties** window of the **Amount** field displays.
- In the **FieldClass** property field, select **FlowField** from the drop-down list.
- In the **CalcFormula** property field, enter the calculation formula for the FlowField.
- (optional) In the **Editable** field, select **No**.

Also the SumIndexField has to be associated with a key



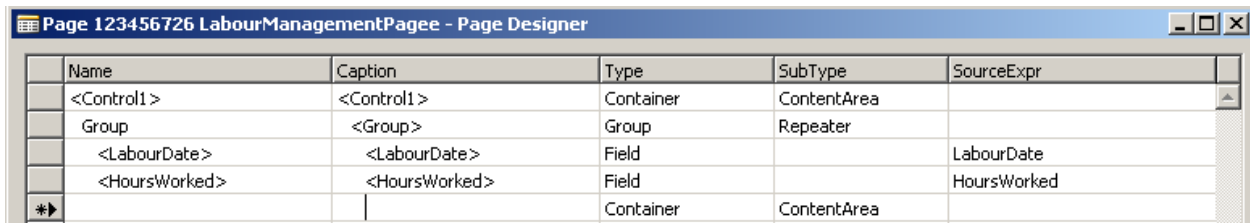
Picture 2: Key define

Then the calcformula can be entered



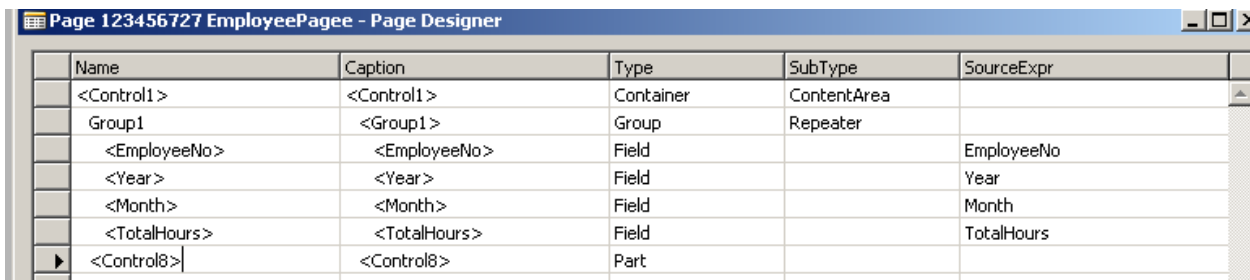
Picture 3: entering calcformula

### 3.3. CREATING A PAGE WITH A EMBEDDED LIST PART



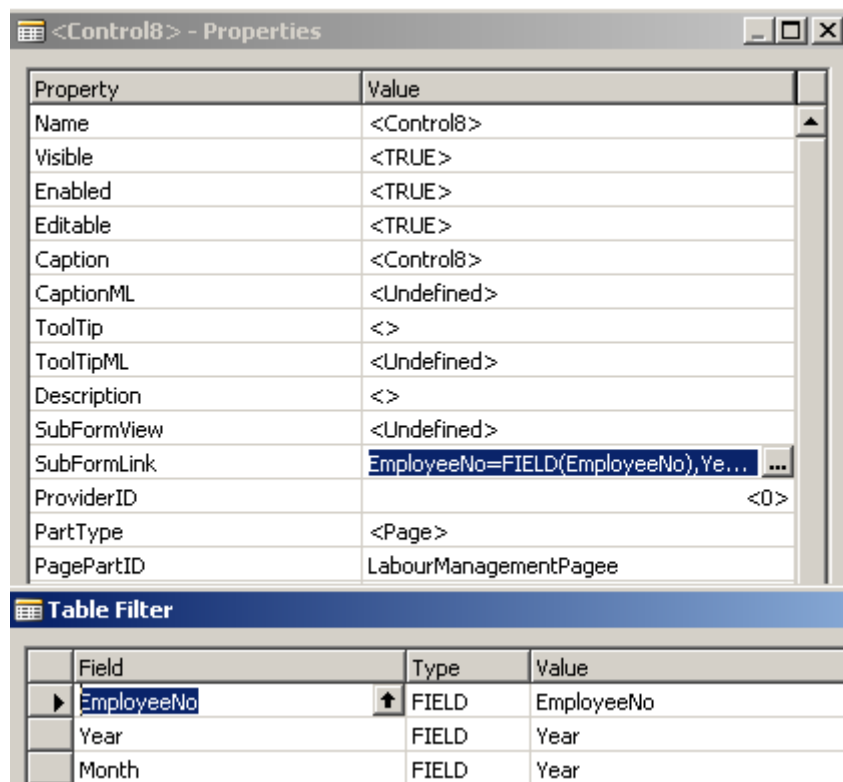
Name	Caption	Type	SubType	SourceExpr
<Control1>	<Control1>	Container	ContentArea	
Group	<Group>	Group	Repeater	
<LabourDate>	<LabourDate>	Field		LabourDate
<HoursWorked>	<HoursWorked>	Field		HoursWorked
*>		Container	ContentArea	

Picture 4: Page creation Labour



Name	Caption	Type	SubType	SourceExpr
<Control1>	<Control1>	Container	ContentArea	
Group1	<Group1>	Group	Repeater	
<EmployeeNo>	<EmployeeNo>	Field		EmployeeNo
<Year>	<Year>	Field		Year
<Month>	<Month>	Field		Month
<TotalHours>	<TotalHours>	Field		TotalHours
>  <Control8>	<Control8>	Part		

Picture 5: Page creation Employee



Property	Value
Name	<Control8>
Visible	<TRUE>
Enabled	<TRUE>
Editable	<TRUE>
Caption	<Control8>
CaptionML	<Undefined>
ToolTip	<>
ToolTipML	<Undefined>
Description	<>
SubFormView	<Undefined>
SubFormLink	EmployeeNo=FIELD(EmployeeNo),Ye...
ProviderID	<0>
PartType	<Page>
PagePartID	LabourManagementPagee

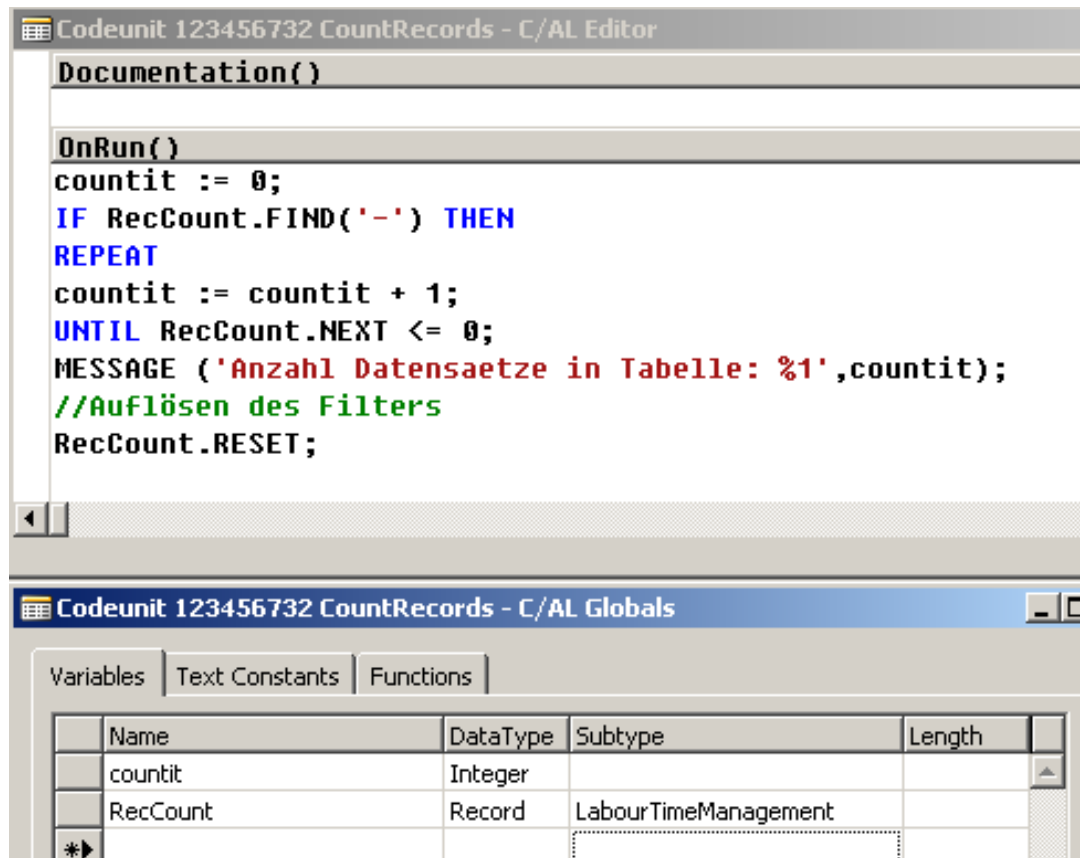
Table Filter			
	Field	Type	Value
>	EmployeeNo	FIELD	EmployeeNo
	Year	FIELD	Year
	Month	FIELD	Year

Picture 6: setting properties

Or without Year and Month? Not working. Only works with single EmployeeNo as Link.

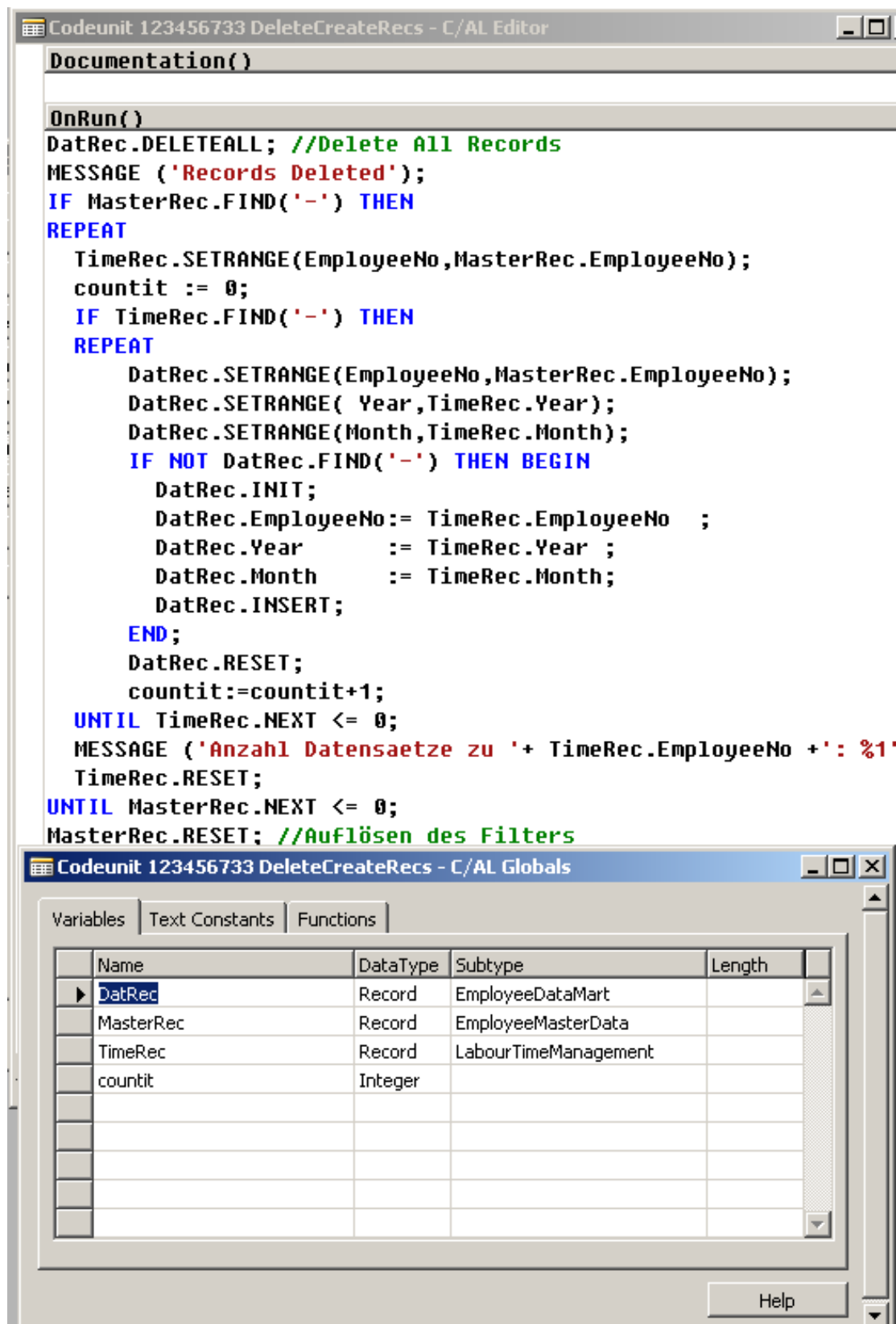
## 4. CODEUNITS

### 4.1. CODEUNIT THAT COUNTS RECORDS



Picture 7: Source Code for Count Records

## 4.2. CODEUNIT THAT DELETES AND CREATES RECORDS



The screenshot shows two windows from the SAP AL Editor. The top window, titled 'Codeunit 123456733 DeleteCreateRecs - C/AL Editor', displays the following source code:

```

Documentation()

OnRun()
DatRec.DELETEALL; //Delete All Records
MESSAGE ('Records Deleted');
IF MasterRec.FIND('-') THEN
REPEAT
  TimeRec.SETRANGE(EmployeeNo,MasterRec.EmployeeNo);
  countit := 0;
  IF TimeRec.FIND('-') THEN
  REPEAT
    DatRec.SETRANGE(EmployeeNo,MasterRec.EmployeeNo);
    DatRec.SETRANGE( Year,TimeRec.Year);
    DatRec.SETRANGE(Month,TimeRec.Month);
    IF NOT DatRec.FIND('-') THEN BEGIN
      DatRec.INIT;
      DatRec.EmployeeNo:= TimeRec.EmployeeNo ;
      DatRec.Year      := TimeRec.Year ;
      DatRec.Month     := TimeRec.Month;
      DatRec.INSERT;
    END;
    DatRec.RESET;
    countit:=countit+1;
  UNTIL TimeRec.NEXT <= 0;
  MESSAGE ('Anzahl Datensätze zu '+ TimeRec.EmployeeNo + ': %1'
    TimeRec.RESET;
UNTIL MasterRec.NEXT <= 0;
MasterRec.RESET; //Auflösen des Filters
  
```

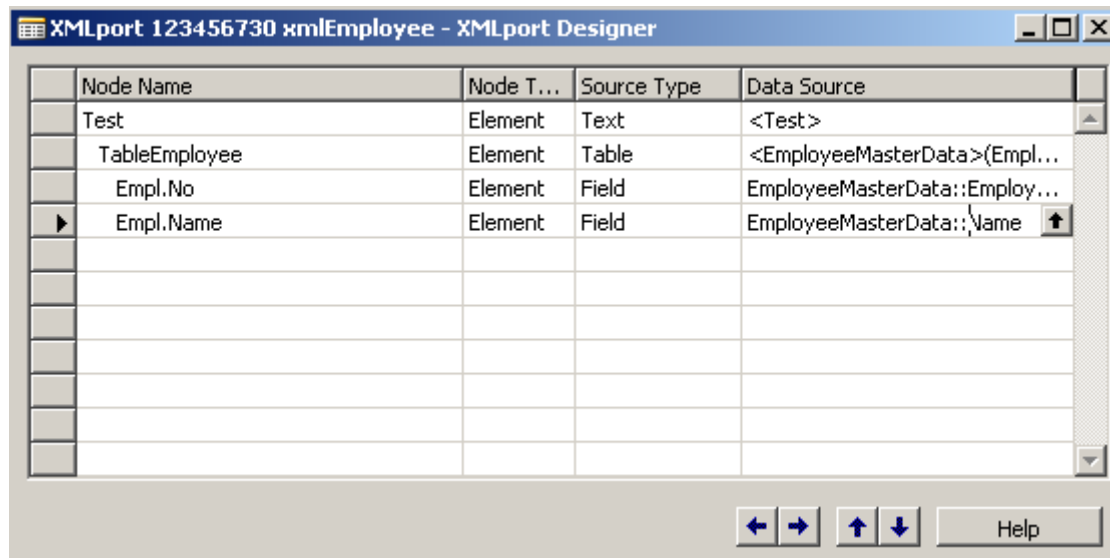
The bottom window, titled 'Codeunit 123456733 DeleteCreateRecs - C/AL Globals', shows the global variables defined for this codeunit:

Name	Data Type	Subtype	Length
DatRec	Record	EmployeeDataMart	
MasterRec	Record	EmployeeMasterData	
TimeRec	Record	LabourTimeManagement	
countit	Integer		

Picture 8: Source Code for delete and create records

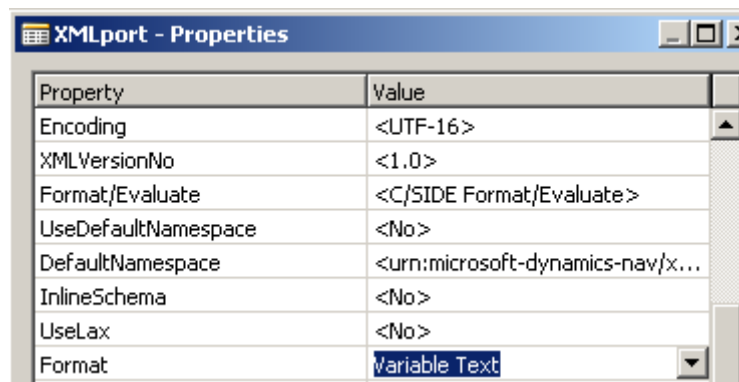
## 5. WRITING TO A FILE WITH XML PORT

### 5.1. Create a XMLport as follows:



Picture 9: xml port Designer

### 5.2. Do the following change in the properties of the XMLport.

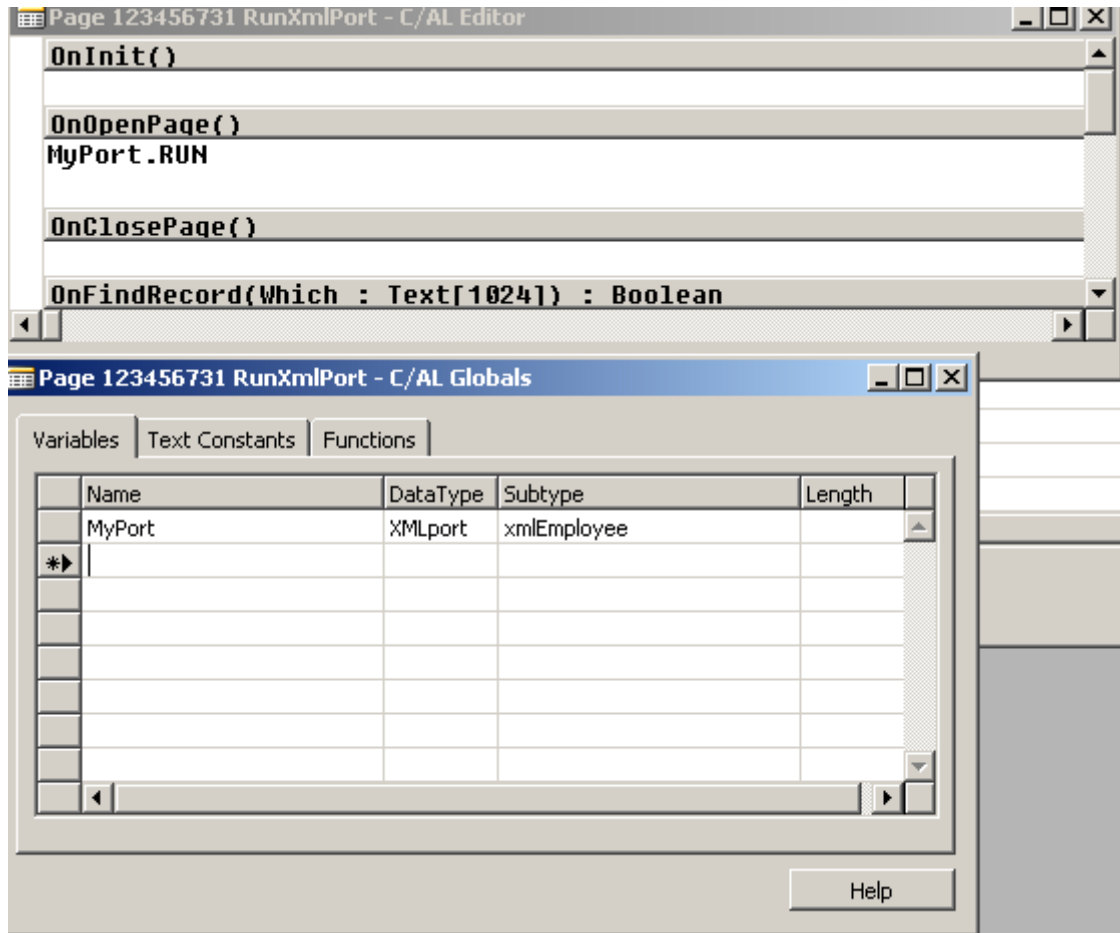


Picture 10: Set properties

This is only needed if a text file is wanted instead of an xml file.

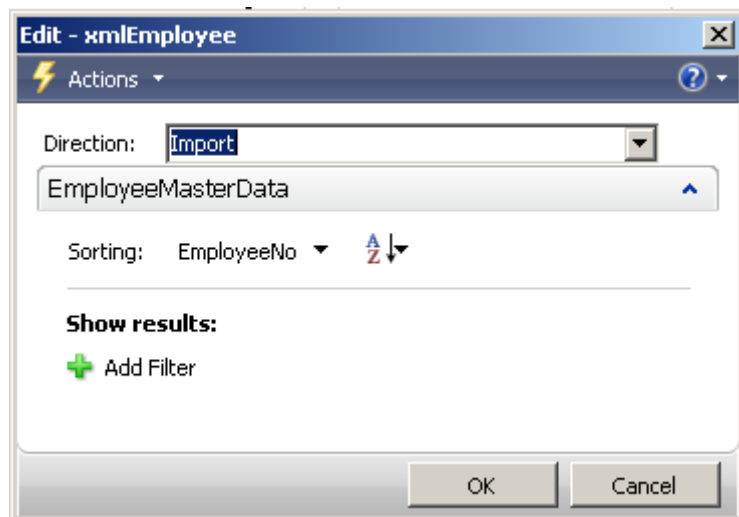


### 5.3. Create a unbounded, blank page and edit Globals and Code:



Picture 11: Source Code for XML

### 5.4. Running the page lets the user now chose import/export of text files.



Picture 12: Created Page for Import/Export