

-----  
'KOCH MEDIA Vertriebsdatenbank

'IMPORT-MODUL

'von Andreas Zahnleiter, Mai 2008

'Akademische Arbeitsgemeinschaft Verlag

'Wolters Kluwer Deutschland GmbH

'Janderstr. 10

'68199 Mannheim  
-----

Option Compare Database

Dim opt As Integer 'Ausgewählter Report-Typ (1=Channel, 2=Stock and Sales, -1=ungültige Datei)

Dim objExcel

Dim objWorkbook

Dim Conn As New ADODB.Connection

Dim DBS As New ADODB.Recordset

Private Sub importKochMediaReport(fileName As String)

Dim num\_rows As Integer 'Anzahl Datensätze

'Channel Report Variablen

Dim lvl As Integer 'Channel Level (1=OVERVIEW, 2=LEVEL1, 3=LEVEL2)

Dim period As String 'Periode des Channel Reports

Dim article\_desc As String

Dim channel\_desc As String

Dim num\_channels As Integer 'Anzahl Channels

Dim channel\_report\_id As Integer

Dim product\_id As Integer

Dim channel\_id As Integer

Dim quantitiy As Long

'Stock and Sales Variablen

Dim month As String

Dim price As String

Dim first\_stock\_date As String

Dim released\_date As String

Dim item\_status As String

Dim status\_id As Integer

Dim stocksales\_report\_id As Long

Dim export\_date As String

Dim export\_time As String

Dim i, j

'Excel Datei öffnen

Set objExcel = CreateObject("Excel.Application")

objExcel.Visible = True

Set objWorkbook = objExcel.Workbooks.Open(fileName)

'Ermittle um welchen Report-Typ es sich hier handelt

'Steht im Feld A1 'month', dann handelt es sich um ein Stock and Sales Report

'ansonsten um ein Channel Report

```
If objWorkbook.Worksheets(1).Range("A1").Value = "month" Then
    opt = 2
ElseIf objWorkbook.Worksheets(1).Range("A2").Value = "company" Then
    opt = 1
Else
    opt = -1

    objExcel.Visible = False
    MsgBox "Fehler! Es handelt sich um keinen Koch Media-Report!", vbCritical
End If
```

'Channel-Report importieren

```
If opt = 1 Then
    objWorkbook.Worksheets("Deutschland").Activate

    'Channel Level ermitteln
    txt = objExcel.ActiveSheet.Range("J1").Value

    If InStr(1, txt, "xx", vbTextCompare) = 0 Then
        If InStr(1, txt, "x", vbTextCompare) = 0 Then
            lvl = 3
        Else
            lvl = 2
        End If
    Else
        lvl = 1
    End If
```

'Anzahl Datensätze ermitteln

```
num_rows = 0
objExcel.ActiveSheet.Range("A3").Select

Do While objExcel.ActiveCell.Value <> ""
    num_rows = num_rows + 1
    objExcel.ActiveCell.Offset(1, 0).Select
Loop
```

'Anzahl Channels ermitteln

```
num_channels = 0
objExcel.ActiveSheet.Range("J1").Select

Do While objExcel.ActiveCell.Value <> ""
    num_channels = num_channels + 1
    objExcel.ActiveCell.Offset(0, 1).Select
Loop
```

'Periode ermitteln

```
period = objExcel.ActiveSheet.Range("D3").Value
```

'Überprüfen ob der Report schon einmal importiert wurde

```
Set Conn = CurrentProject.Connection
Set DBS = New ADODB.Recordset
```

```
DBS.Open "SELECT * FROM tbl_channel_report WHERE tbl_channel_report.period = '" & period  
& "' AND tbl_channel_report.channel_level_id = '" & lvl, Conn, adOpenKeyset, adLockOptimistic
```

```
If DBS.RecordCount = 0 Then
```

```
'Nein, Report noch nicht vorhanden
```

```
'Datensatz in die Tabelle tbl_channel_report anfügen
```

```
DBS.Close
```

```
DBS.Open "tbl_channel_report", Conn, adOpenKeyset, adLockOptimistic
```

```
DBS.AddNew
```

```
DBS!channel_level_id = lvl
```

```
DBS!period = period
```

```
DBS!import_date = Now()
```

```
DBS.Update
```

```
DBS.Close
```

```
'Überprüfen, ob alle Artikel schon vorhanden sind, wenn nein, dann
```

```
'fehlende Artikeln importieren
```

```
objExcel.ActiveSheet.Range("H3").Select
```

```
i = 1
```

```
'-----
```

```
'Alle Artikeln durchlaufen
```

```
Do
```

```
    article_desc = objExcel.ActiveCell.Value
```

```
    DBS.Open "SELECT * FROM tbl_product WHERE tbl_product.description = '" &  
article_desc & "'", Conn, adOpenKeyset, adLockOptimistic
```

```
If DBS.RecordCount = 0 Then
```

```
'Artikel noch nicht vorhanden, importieren
```

```
DBS.Close
```

```
DBS.Open "tbl_product", Conn, adOpenKeyset, adLockOptimistic
```

```
DBS.AddNew
```

```
DBS!Description = article_desc
```

```
DBS.Update
```

```
DBS.Close
```

```
Else
```

```
'Artikel schon vorhanden
```

```
DBS.Close
```

```
End If
```

```
objExcel.ActiveCell.offset(1, 0).Select
```

```
i = i + 1
```

```
Loop While i <= num_rows
```

```
'Überprüfen, ob alle Channels importiert wurden, wenn nein, dann fehlende
```

```
'Channels importieren
```

```
objExcel.ActiveSheet.Range("J1").Select
```

```
i = 1

'-----
'Alle Channels durchlaufen
Do
    channel_desc = objExcel.ActiveCell.Value

    DBS.Open "SELECT * FROM tbl_channel WHERE tbl_channel.description = '" &
channel_desc & "'", Conn, adOpenKeyset, adLockOptimistic

    If DBS.RecordCount = 0 Then
        'Channel noch nicht vorhanden, importieren
        DBS.Close

        DBS.Open "tbl_channel", Conn, adOpenKeyset, adLockOptimistic

        DBS.AddNew
        DBS!Description = channel_desc
        DBS.Update

        DBS.Close
    Else
        'Channel schon vorhanden
        DBS.Close
    End If

    objExcel.ActiveCell.offset(0, 1).Select
    i = i + 1
Loop While i <= num_channels

'-----
'Zahlen importieren
i = 1

Do 'Alle Artikel durchlaufen

    objExcel.ActiveSheet.Range("H2").Select
    objExcel.ActiveCell.offset(i, 0).Select

    'product_id auslesen
    DBS.Open "SELECT tbl_product.id FROM tbl_product WHERE tbl_product.description =
'" & objExcel.ActiveCell.Value & "'", Conn, adOpenKeyset, adLockOptimistic
    product_id = DBS!Id
    DBS.Close

    'channel_report_id auslesen
    DBS.Open "SELECT tbl_channel_report.id FROM tbl_channel_report WHERE
tbl_channel_report.period = '" & period & "' AND tbl_channel_report.channel_level_id = '" &
lvl, Conn, adOpenKeyset, adLockOptimistic
    channel_report_id = DBS!Id
    DBS.Close

    j = 1

Do 'Alle Channels durchlaufen

    objExcel.ActiveSheet.Range("H2").Select
```

```

objExcel.ActiveCell.Offset(i, 0).Select

'channel_id auslesen
objExcel.ActiveCell.Offset(-1 - i, 1 + j).Select
DBS.Open "SELECT tbl_channel.id FROM tbl_channel WHERE
tbl_channel.description = '" & objExcel.ActiveCell.Value & "'", Conn, adOpenKeyset,
adLockOptimistic
channel_id = DBS!Id
DBS.Close

'quantity auslesen
quantity = ReadDataFromExcel(1 + i, 0)

'tbl_report_product_channel Datensatz anlegen
DBS.Open "tbl_report_product_channel", Conn, adOpenKeyset, adLockOptimistic
DBS.AddNew
DBS!channel_report_id = channel_report_id
DBS!product_id = product_id
DBS!channel_id = channel_id
DBS!quantity = quantity
DBS.Update

DBS.Close

j = j + 1

Loop While j <= num_channels

i = i + 1

Loop While i <= num_rows

'Excel Datei wieder schließen
objWorkbook.Close True
objExcel.Quit

MsgBox "Der Bericht wurde erfolgreich importiert!", vbInformation

Else
'Excel Datei wieder schließen
objWorkbook.Close True
objExcel.Quit

MsgBox "Der Bericht wurde schon importiert!", vbCritical
DBS.Close
End If

'-----
'Stock and Sales Report importieren
ElseIf opt = 2 Then

objWorkbook.Worksheets(1).Activate

'Anzahl Datensätze ermitteln
num_rows = 0
objExcel.ActiveSheet.Range("A2").Select

```

```
Do While objExcel.ActiveCell.Value <> ""
    num_rows = num_rows + 1
    objExcel.ActiveCell.Offset(1, 0).Select
Loop

'Monat ermitteln
month = objExcel.ActiveSheet.Range("A2").Value

'export_date ermitteln
export_date = objExcel.ActiveSheet.Range("AF2").Value

'export_time ermitteln
export_time = objExcel.ActiveSheet.Range("AG2").Value

'Überprüfen ob der Report schon einmal importiert wurde
Set Conn = CurrentProject.Connection
Set DBS = New ADODB.Recordset

DBS.Open "SELECT * FROM tbl_stocksales_report WHERE tbl_stocksales_report.month_report = " & month & "", Conn, adOpenKeyset, adLockOptimistic

If DBS.RecordCount = 0 Then
    'Nein, Report noch nicht vorhanden

    'Datensatz in die Tabelle tbl_channel_report anfügen
    DBS.Close

    DBS.Open "tbl_stocksales_report", Conn, adOpenKeyset, adLockOptimistic

    DBS.AddNew
    DBS!month_report = month
    DBS!import_date = Now()
    DBS!export_date = export_date
    DBS!export_time = export_time
    DBS.Update

    DBS.Close

    '-----

    'Überprüfen, ob alle Artikel schon vorhanden sind, wenn nein, dann
    'fehlende Artikeln importieren
    objExcel.ActiveSheet.Range("E2").Select

    i = 1

    'Alle Artikel durchlaufen
    Do
        article_desc = objExcel.ActiveCell.Value

        DBS.Open "SELECT * FROM tbl_product WHERE tbl_product.description = " & article_desc & "", Conn, adOpenKeyset, adLockOptimistic

        If DBS.RecordCount = 0 Then
            'Artikel noch nicht vorhanden, importieren
            DBS.Close
```

```
DBS.Open "tbl_product", Conn, adOpenKeyset, adLockOptimistic

DBS.AddNew
DBS!Description = article_desc
DBS.Update

DBS.Close
Else
    'Artikel schon vorhanden
    DBS.Close
End If

objExcel.ActiveCell.offset(1, 0).Select
i = i + 1
Loop While i <= num_rows

'-----

'Überprüfen, ob alle Statusmöglichkeiten schon vorhanden sind, wenn nein, dann
'fehlende importieren
objExcel.ActiveSheet.Range("I2").Select

i = 1

'Alle Datensätze durchlaufen
Do
    item_status = objExcel.ActiveCell.Value

    DBS.Open "SELECT * FROM tbl_product_status WHERE tbl_product_status.description
= '" & item_status & "'", Conn, adOpenKeyset, adLockOptimistic

    If DBS.RecordCount = 0 Then
        'Status noch nicht vorhanden, importieren
        DBS.Close

        DBS.Open "tbl_product_status", Conn, adOpenKeyset, adLockOptimistic

        DBS.AddNew
        DBS!Description = item_status
        DBS.Update

        DBS.Close
    Else
        'Artikel schon vorhanden
        DBS.Close
    End If

    objExcel.ActiveCell.offset(1, 0).Select
    i = i + 1
Loop While i <= num_rows

'-----

'Überprüfen, ob noch fehlende Artikelinformationen hinzugefügt werden müssen
i = 1

'Alle Artikeln durchlaufen
```

Do

objExcel.ActiveSheet.Range("E1").Select

article\_desc = ReadDataFromExcel(i, 0)

DBS.Open "SELECT \* FROM tbl\_product WHERE tbl\_product.description = '" &  
article\_desc & "'", Conn, adOpenKeyset, adLockOptimistic

If DBS.RecordCount <> 0 Then

DBS.Close

'Preis auslesen

price = ReadDataFromExcel(0, 5)

'1st stock date auslesen

first\_stock\_date = ReadDataFromExcel(0, 1)

'released date auslesen

released\_date = ReadDataFromExcel(0, 1)

DBS.Open "SELECT \* FROM tbl\_product WHERE tbl\_product.description = '" &  
article\_desc & "'", Conn, adOpenKeyset, adLockOptimistic

DBS!price = price

DBS!first\_stock\_date = first\_stock\_date

DBS!release\_date = released\_date

DBS.Update

DBS.Close

Else

'Fehler

DBS.Close

End If

i = i + 1

Loop While i <= num\_rows

'-----

'Zahlen importieren

i = 1

Do 'Alle Artikel durchlaufen

objExcel.ActiveSheet.Range("E1").Select

objExcel.ActiveCell.offset(i, 0).Select

'product\_id auslesen

DBS.Open "SELECT tbl\_product.id FROM tbl\_product WHERE tbl\_product.description =  
'" & objExcel.ActiveCell.Value & "'", Conn, adOpenKeyset, adLockOptimistic

product\_id = DBS!Id

DBS.Close

'item status auslesen

item\_status = ReadDataFromExcel(0, 4)

'status\_id ermitteln

DBS.Open "SELECT tbl\_product\_status.id FROM tbl\_product\_status WHERE



```

tbl_product_status.description = ' ' & item_status & ' ', Conn, adOpenKeyset, adLockOptimistic
    status_id = DBS!Id
    DBS.Close

    'stocksales_report_id ermitteln
    DBS.Open "SELECT tbl_stocksales_report.id FROM tbl_stocksales_report WHERE
tbl_stocksales_report.month_report = ' ' & month & ' ', Conn, adOpenKeyset, adLockOptimistic
    stocksales_report_id = DBS!Id
    DBS.Close

    'tbl_product_stocksales Datensatz anlegen
    DBS.Open "tbl_product_stocksales", Conn, adOpenKeyset, adLockOptimistic
    DBS.AddNew
    DBS!product_id = product_id
    DBS!stocksales_report_id = stocksales_report_id
    DBS!status_id = status_id
    DBS!stock_entry_lifetime = ReadDataFromExcel(0, 4)
    DBS!customer_orders_1 = ReadDataFromExcel(0, 1)
    DBS!customer_orders_2 = ReadDataFromExcel(0, 1)
    DBS!customer_orders_3 = ReadDataFromExcel(0, 1)
    DBS!sales_lifetime = ReadDataFromExcel(0, 1)
    DBS!returns_lifetime = ReadDataFromExcel(0, 1)
    DBS!net_sales_prev_year = ReadDataFromExcel(0, 1)
    DBS!net_sales_curr_year = ReadDataFromExcel(0, 1)
    DBS!net_sales_1 = ReadDataFromExcel(0, 1)
    DBS!net_sales_2 = ReadDataFromExcel(0, 1)
    DBS!net_sales_3 = ReadDataFromExcel(0, 1)
    DBS!net_sales_4 = ReadDataFromExcel(0, 1)
    DBS!backorders_customers = ReadDataFromExcel(0, 1)
    DBS!stock_on_hand = ReadDataFromExcel(0, 1)
    DBS!external_stock = ReadDataFromExcel(0, 1)
    DBS!commission_stock = ReadDataFromExcel(0, 1)
    DBS!total_stock = ReadDataFromExcel(0, 1)
    DBS!backorder_vendor = ReadDataFromExcel(0, 1)
    DBS!delivery_date = ReadDataFromExcel(0, 1)
    DBS.Update

    DBS.Close

    i = i + 1

Loop While i <= num_rows

'Excel Datei wieder schließen
objWorkbook.Close True
objExcel.Quit

Else
    DBS.Close

    DBS.Open "SELECT * FROM tbl_stocksales_report WHERE
tbl_stocksales_report.month_report = ' ' & month & ' ', Conn, adOpenKeyset, adLockOptimistic

    objExcel.Visible = False

    If MsgBox("Es liegt schon ein Bericht vom " & month & " mit Export-Datum " &
DBS!export_date & " " & DBS!export_time & " vor! Wollen Sie die vorhandenen Daten

```

überschreiben?", vbYesNo, "Hinweis: Daten-Import") = vbYes Then

DBS.Close

'Bericht aktualisieren

'Tabelle tbl\_stocksales\_report aktualisieren

DBS.Open "SELECT \* FROM tbl\_stocksales\_report WHERE

tbl\_stocksales\_report.month\_report=' ' & month & ' ', Conn, adOpenKeyset, adLockOptimistic

DBS!import\_date = Now()

DBS!export\_date = export\_date

DBS!export\_time = export\_time

DBS.Update

DBS.Close

objExcel.Visible = True

'-----

'Zahlen aktualisieren

'stocksales\_report\_id ermitteln

DBS.Open "SELECT tbl\_stocksales\_report.id FROM tbl\_stocksales\_report WHERE

tbl\_stocksales\_report.month\_report = ' ' & month & ' ', Conn, adOpenKeyset, adLockOptimistic

stocksales\_report\_id = DBS!Id

DBS.Close

i = 1

Do 'Alle Artikel durchlaufen

objExcel.ActiveSheet.Range("E1").Select

objExcel.ActiveCell.Offset(i, 0).Select

'product\_id auslesen

DBS.Open "SELECT tbl\_product.id FROM tbl\_product WHERE

tbl\_product.description = ' ' & objExcel.ActiveCell.Value & ' ', Conn, adOpenKeyset, adLockOptimistic

product\_id = DBS!Id

DBS.Close

'item status auslesen

item\_status = ReadDataFromExcel(0, 4)

'status\_id ermitteln

DBS.Open "SELECT tbl\_product\_status.id FROM tbl\_product\_status WHERE

tbl\_product\_status.description = ' ' & item\_status & ' ', Conn, adOpenKeyset, adLockOptimistic

status\_id = DBS!Id

DBS.Close

'tbl\_product\_stocksales Datensatz aktualisieren

DBS.Open "SELECT \* FROM tbl\_product\_stocksales WHERE

tbl\_product\_stocksales.stocksales\_report\_id=" & stocksales\_report\_id & " AND

tbl\_product\_stocksales.product\_id=" & product\_id, Conn, adOpenKeyset, adLockOptimistic

DBS!status\_id = status\_id

DBS!stock\_entry\_lifetime = ReadDataFromExcel(0, 4)

```

DBS!customer_orders_1 = ReadDataFromExcel(0, 1)
DBS!customer_orders_2 = ReadDataFromExcel(0, 1)
DBS!customer_orders_3 = ReadDataFromExcel(0, 1)
DBS!sales_lifetime = ReadDataFromExcel(0, 1)
DBS!returns_lifetime = ReadDataFromExcel(0, 1)
DBS!net_sales_prev_year = ReadDataFromExcel(0, 1)
DBS!net_sales_curr_year = ReadDataFromExcel(0, 1)
DBS!net_sales_1 = ReadDataFromExcel(0, 1)
DBS!net_sales_2 = ReadDataFromExcel(0, 1)
DBS!net_sales_3 = ReadDataFromExcel(0, 1)
DBS!net_sales_4 = ReadDataFromExcel(0, 1)
DBS!backorders_customers = ReadDataFromExcel(0, 1)
DBS!stock_on_hand = ReadDataFromExcel(0, 1)
DBS!external_stock = ReadDataFromExcel(0, 1)
DBS!commission_stock = ReadDataFromExcel(0, 1)
DBS!total_stock = ReadDataFromExcel(0, 1)
DBS!backorder_vendor = ReadDataFromExcel(0, 1)
DBS!delivery_date = ReadDataFromExcel(0, 1)
DBS.Update

```

```

DBS.Close

```

```

i = i + 1

```

```

Loop While i <= num_rows

```

```

End If

```

```

'Excel Datei wieder schließen

```

```

objWorkbook.Close True

```

```

objExcel.Quit

```

```

End If

```

```

'Fehler

```

```

Else

```

```

End If

```

```

Set Conn = Nothing

```

```

Set DBS = Nothing

```

```

Set objWorkbook = Nothing

```

```

Set objExcel = Nothing

```

```

End Sub

```

```

Private Function ReadDataFromExcel(ByVal offset_row As Integer, ByVal offset_col As Integer)

```

```

    objExcel.ActiveCell.offset(offset_row, offset_col).Select

```

```

    ReadDataFromExcel = objExcel.ActiveCell.Value

```

```

End Function

```

```

Private Sub cmd_analyze_data_Click()

```

```

    DoCmd.OpenForm "frm_reporting", acNormal, , , acFormEdit, acWindowNormal

```

```

End Sub

```

```

Private Sub cmd_import_newdata_Click()

```

```

Dim fd As FileDialog
Set fd = Application.FileDialog(msoFileDialogFilePicker)
Dim vrtSelectedItem As Variant

fd.Filters.Clear
fd.Filters.Add "Excel Dateien", "*.xls"

With fd
    If .Show = -1 Then

        For Each vrtSelectedItem In .SelectedItems
            importKochMediaReport (vrtSelectedItem)
        Next vrtSelectedItem

    Else
        End If
End With

Set fd = Nothing

End Sub

Private Sub cmd_quit_Click()
    DoCmd.Quit
End Sub

'-----
'KOCH MEDIA Vertriebsdatenbank
'
'REPORTING-MODUL
'
'von Andreas Zahnleiter, Mai 2008
'
'Akademische Arbeitsgemeinschaft Verlag
'Wolters Kluwer Deutschland GmbH
'Janderstr. 10
'68199 Mannheim
'-----

Option Compare Database

Dim report_type As Integer '1=Pivot 2=Pivot Chart

Private Sub change_subform(name_of_subform As String)
    If frm_reporting_subform.SourceObject <> name_of_subform Then
        frm_reporting_subform.SourceObject = name_of_subform
    End If
End Sub

Private Sub cmd_reporting_close_Click()
    DoCmd.Close acForm, "frm_reporting", acSaveYes
End Sub

Private Sub cmd_reporting_printpreview_Click()
    If report_type = 1 Then

```

```
        DoCmd.OpenForm frm_reporting_subform.SourceObject, acFormPivotTable
Else
        DoCmd.OpenForm frm_reporting_subform.SourceObject, acFormPivotChart
End If
End Sub

Private Sub cmd_show_channels_pivot_Click()
    report_type = 1
    change_subform ("frm_channels_pivot")
End Sub

Private Sub cmd_show_channels_pivotchart_Click()
    report_type = 2
    change_subform ("frm_channels_pivotchart")
End Sub

Private Sub cmd_show_netsales_pivot_Click()
    report_type = 1
    change_subform ("frm_netsales_pivot")
End Sub

Private Sub cmd_show_netsales_pivotchart_Click()
    report_type = 2
    change_subform ("frm_netsales_pivotchart")
End Sub

Private Sub cmd_show_stockonhand_pivot_Click()
    report_type = 1
    change_subform ("frm_stockonhand_pivot")
End Sub

Private Sub cmd_show_stockonhand_pivotchart_Click()
    report_type = 2
    change_subform ("frm_stockonhand_pivotchart")
End Sub

Private Sub Form_Load()

End Sub
```