



Access cursus - Hoofdstuk 19

Handleiding van Helpmij.nl

Auteur: OctaFish

Augustus 2013

“ Dé grootste en gratis computerhelpdesk van Nederland ”

Access cursus - Hoofdstuk 19

Office Automation

Office gebruikers werken meestal binnen één programma, om de simpele reden dat het gekozen programma prima voldoet aan de eisen die de gebruiker op dat moment aan zijn gereedschap stelt. Zo voldoet Word uitstekend als je een brief wilt maken, en is Excel uitstekend in grafieken. Voor een eenvoudige tabel kun je ook best uit de voeten met Word, maar gaat het om een adressenbestand van een paar honderd adressen, dan zul je dat niet zo snel in Word doen, maar eerder naar Excel stappen. En gaat het om een adressenbestand voor een vereniging van een paar duizend leden, dan kun je misschien beter overstappen naar Access. Kortom: net zoals een timmerman, grijp je voor je werkzaamheden naar het juiste gereedschap.

Soms echter is het handiger om de verschillende pakketten te combineren, en zo het eindresultaat te optimaliseren. Je kunt bijvoorbeeld in Access prima rapporten maken voor facturen, maar het aanpassen van de gegenereerde factuur is nog niet zo eenvoudig. Om te beginnen ligt het Access rapport redelijk vast qua lay-out. Wil je dat veranderen, of wil je tekst toevoegen, dan moet je daar behoorlijk wat werk voor verzetten. Het rapport exporteren naar Word is dan eigenlijk geen optie, omdat de opmaak van geconverteerde rapport doorgaans niet bruikbaar is: je factuur kun je dan eigenlijk van voren af aan opnieuw opmaken.

Zou je een factuur in Word maken, dan is het aanpassen veel simpeler, omdat Word nu eenmaal is gebouwd als tekstverwerker. Maar het samenvoegen van de gegevens is dan weer een groot probleem.

Je zou dus voor het genereren van de factuur Access willen gebruiken, omdat je dan alle factuurgegevens netjes kunt verzamelen, maar voor het opmaken van de factuur verder willen werken in Word. Gelukkig kan dit in de Office: het proces dat we hierbij gebruiken noemen we *automation*. Je zult in het proces meestal wel het e.e.a. moeten programmeren, omdat de standaard functionaliteiten van de gebruikte programma's nu eenmaal te kort schiet voor de wat ingewikkelder vormen van samenwerking.

In dit hoofdstuk gaan we een rapport maken in Word dat wordt opgebouwd met een Excel Samenvoegbestand dat is gegenereerd door Access. Het Excel bestand komt uit een query die records van een op te geven aantal dagen bevat, en bevat voor elk Word document andere gegevens (op basis van een klant). De documenten die we gaan genereren worden opgeslagen onder de naam van de klant; we moeten die naam dus genereren in het proces.

Het gebruikte voorbeeld kan vermoedelijk ook wel op een simpelere manier bereikt worden, maar het gaat er nu om dat de verschillende technieken gedemonstreerd worden; de praktische toepassingen ga je uiteraard zelf verzinnen!

De oefening maakt gebruik van een aantal gegevensbronnen:

- Access: tabel *tUitleveren*
- Word: Samenvoegbestand *Merge.doc*
- Excel bestand *UitleverMerge*

De tabel in Access wordt automatisch aangemaakt in de procedure, dus die hoeft je niet zelf aan te maken. Het Word bestand is een samenvoegbestand dat de gegevens uit het Excel bestand haalt. Beide bestanden moeten dus bestaan, en voor de beste werking zou je het Word bestand al een keer met de hand moeten samenvoegen, zodat je de gegevens op de juiste plaats kunt zetten. Als de koppeling eenmaal bestaat, kan het Excel bestand probleemloos worden overschreven, en dat is dan ook precies wat we gaan doen.

Zoals je ondertussen gewend bent, breken we de code weer in hapklare brokken, en bekijken we elk

stuk op zijn relevante werking. Allereerst krijgt de functie een naam, en beginnen we met het vastleggen van de variabelen die we gaan gebruiken.

```
Private Function BedrijfMail()
Dim aWord As Object, oWord As Word.Document
Dim strPad as String, strMergePad As String, strMergeDoc As
String
Dim rst As ADODB.Recordset
Dim qTmp As QueryDef
Dim strVP As Variant, temp as Variant
Dim strSQL as String, sBedrijf As String

strPad = CurrentProject.Path & "Dagelijks"           'Pad voor
uitleveringen
strMergePad = CurrentProject.Path & "Merge"         'Document
voor emailmerge
strMergeDoc = strMergePad & "Merge.doc"

strSQL = "Select * From tUitleveren"
```

Nu de variabelen zijn gedefinieerd en gevuld, kunnen we aan de slag. De tabel [tUitleveren] bevat de records die we gaan gebruiken voor het Word rapport. Die tabel moet eerst gemaakt worden. We gebruiken een aparte tabel voor de export, en niet een query (wat op zich prima zou kunnen) maar omdat we een samenvoeging gaan maken waarbij de bedrijfsnaam onderdeel is van de rapportnaam, moeten we er zeker van zijn dat de bedrijfsnaam een geldige bestandsnaam gaat opleveren. Een bedrijf als Octa/Foto gaat niet werken, omdat een /-teken niet is toegestaan in een bestandsnaam. We moeten de bedrijfsnamen dus checken, en zondig aanpassen. Dat wil je uiteraard niet doen in de hoofdtabellen, maar een tijdelijke tabel is daar prima geschikt voor.

```
If WeekDay(date, vbMonday) = 1 Or WeekDay(date, vbMonday) = 2
Then iAantal
= 4 Else iAantal = 2
On Error GoTo Stoppen
iAantalDagen = InputBox("Hoeveel dagen geleden?",
"Contactdatum", iAantal)
On Error GoTo 0

DoCmd.Echo False, "Eerst de uitlevertabel maken..."
DoCmd.SetWarnings False
On Error Resume Next
CurrentDb.TableDefs.Delete ("tUitleveren")
strSQL = "SELECT qWordMerge.* INTO tUitleveren FROM qWordMerge "
& vbCrLf
& "WHERE (CONTACTDATUM = Date()-" & iAantalDagen);"
DoCmd.RunSQL strSQL
DoCmd.SetWarnings True
```

De tabel [tUitleveren] wordt eerst verwijderd; we gaan hem namelijk opnieuw aanmaken. Meestal maak ik een tijdelijke tabel leeg, en vul ik hem met de nieuwe waarden. De reden daarvoor is dat de lay-out van de tabel dan bewaard blijft. In dit geval is dat niet echt nodig, omdat we verder met de tabel weinig doen. Dus vervangen we de tabel.

De query qWordMerge is een query die de gegevens ophaalt die we nodig hebben voor de samenvoeging. Hij bevat de volgende velden:

```
Id, pid, bedrijfsnr, bedrijfsnaam, straat, huisnr, huisnr_toev,
postcode, plaats,

telefoon, bik, bik_omschrijving, kwp, klasse, geslacht,
```

```

voorletters, tussenvoegsels,

achternaam, functie, email, contactdatum, adviseur,
afspraakdatum

```

De hoofdtabel wordt gefilterd op basis van een datumfilter. Dat filter (iAantalDagen) kijkt naar de dag van de week; het heeft weinig zin om op Maandag de records van één dag terug te nemen, want op Zondag zal er weinig gewerkt zijn. Dus op Maandag zul je de records van Vrijdag willen leveren. Al de bovenstaande velden vind je straks dus ook terug in de tabel [tUitleveren].

```

strSQL = "Select * From tUitleveren"
Set rst = New Recordset
rst.Open strSQL, CurrentProject.Connection, adOpenKeyset,
adLockOptimistic
strVP = rst.GetRows

```

De recordset wordt geopend, en ingelezen in de Array variabele strVP. Deze variabele is niet als zodanig gedefinieerd, (dan zou hij er zo uit hebben gezien: Dim strVP() As Variant) maar omdat een variant geen vaste dimensies heeft, kunnen we de variabele vullen met een recordset. Daarbij wordt de matrix ingesteld op het aantal rijen en velden.

Opmerking: een recordset kan veel groter zijn dan er in een matrix variabele past. In dat geval wordt de recordset dus niet volledig gebruikt. Controleer dus of de matrix wel goed gevuld wordt!

In de procedure kijken we eerst of er records in de recordset zitten, anders heeft de procedure weinig zin.

```

If rst.RecordCount > 0 Then

    strFileTxT = strMergePad & "UitleverMerge.xls"
    For i = LBound(strVP) To UBound(strVP)
        sBedrijf = strVP(3, i)
        GoSub BedrijfCheck
        strVP(3, i) = Trim(sBedrijf)
        strUitgeleverd = Trim(strVP(3, i)) & " - Afspraken " &
Date & ".doc"
        temp = Dir(strPad & strUitgeleverd)

```

De lus gebruikt de array strVP die is gevuld met de recordset. LBound(strVP) geeft het eerste recordnummer aan, en UBound(strVP) het laatste record. Zoals je uit de velden uit de query qWordMerge kunt zien, bevat het vierde veld de bedrijfsnaam strVP(3, i), en die zetten we voor elke uitlevering in een stringvariabele (strUitgeleverd). In de subroutine BedrijfCheck wordt de juiste syntax voor de bedrijfsnaam gecontroleerd. Vervolgens kijken we of het bestand al bestaat of niet.

```

If strUitgeleverd <> temp Then

    strSQL = "SELECT * FROM tUitleveren WHERE [ID] =" & strVP(0, i)
    CurrentDb.QueryDefs.Delete ("qUitleverMerge")
    Set qTmp = CurrentDb.CreateQueryDef("qUitleverMerge", strSQL)

```

Voor de query qUitleverMerge geldt hetzelfde als voor de tabel tUitleveren: we zouden de SQL ervan kunnen vervangen door de nieuwe SQL. Omdat de query verder niet gebruikt wordt, en om te voorkomen dat er een check moet worden ingebouwd op het aanwezig zijn van de query, gebruiken we de iets bottere methode van het weggooien van de query, en maken we hem elke keer opnieuw aan.

```

On Error Resume Next

```

```

Kill strFileTxT
  Err.Clear
  On Error GoTo 0
  DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9,

  "tUitleverMerge", strMergePad & "UitleverMerge.xls"

```

Met Kill wordt het oude bestand verwijderd, en daarna gebruiken we TransferSpreadsheet om de tabel tUitleverMerge te exporteren naar het bestand UitleverMerge.xls. Dit wordt, zoals je vast nog weet, gebruikt in de Samenvoeging met Word. En wordt steeds opnieuw vervangen.

```

Set aWord = CreateObject("Word.Application")

Set oWord = aWord.Documents.Open(strMergeDoc)

```

Vervolgens wordt een Word sessie gestart, en in die sessie wordt het Mergedocument geopend. Dit document is zoals gezegd al helemaal voorbereid op samenvoegen met het Excel bestand, dus we hebben daar verder weinig omkijken naar. Het samenvoegen kunnen we dus gelijk opstarten.

```

On Error Resume Next
  With oWord.MailMerge
    .Destination = wdSendToNewDocument
    .Execute
  End With

```

Dat doen we met Word commando's zoals MailMerge en wdSendToNewDocument.

```

'Bestand opslaan
  strUitgeleverd = strPad & Trim(strVP(3, i)) & " -
Afspraken " & Date & ".doc"
  oWord.Application.ActiveDocument.SaveAs (strPad &
strUitgeleverd)
  oWord.Application.ActiveDocument.Close
  oWord.Close (wdDoNotSaveChanges)
  Set oWord = Nothing
  aWord.Application.Quit
End If
Next
End If

```

En we eindigen de Word sessie met het opslaan van het samengevoegde document. Hier wordt de specifieke bestandsnaam afgeleid van de klantnaam die we hebben gehaald uit de matrixvariabele. Daarna kan het document, en de Word sessie worden gesloten.

```

rst.Close
Set rst = Nothing
Set db = Nothing
Set aWord = Nothing
DoCmd.Echo True
On Error Resume Next
aWord.Quit
Set aWord = Nothing
Exit Function

```

'=====

Er is nog wel enige tijdwinst te halen als je één Word sessie start, en die open laat staan gedurende het mergen van de verschillende records. Toen ik deze procedure maakte kwam het een enkele keer voor dat er een sessie open bleef staan, en er een tweede sessie bij werd geopend. Met allerlei problemen (dialogvensters waarop geklikt moet worden) tot gevolg. Voor de zekerheid wordt de Word sessie nu steeds afgesloten, en wordt er steeds een nieuwe geopend. Dat levert een klein beetje tijdverlies op, maar niet overdreven veel, was de ervaring.

'Hieronder staan de subroutine `BedrijfCheck` die boven in de code worden aangesproken. Deze routine controleert de bedrijfsnaam op ongeldige tekens, en vervangt die door aanhalingstekens.

```
'-----
BedrijfCheck:

'routine die controleert of de bedrijfsnaam geschikt is als bestandsnaam.

'Zo niet, dan wordt de Bedrijfsnaam geschoond van niet-toegestane tekens.

'als er een " (=Chr(34)) in de naam voorkomt, vervangen door ' (=Chr(39))

If InStr(1, sBedrijf, Chr(34)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(34), Chr(39), 1)

'als er een (=Chr(92)) in de naam voorkomt, vervangen door '-'

If InStr(1, sBedrijf, Chr(92)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(92), "-", 1)

'als er een / (=Chr(47)) in de naam voorkomt, vervangen door '-'

If InStr(1, sBedrijf, Chr(47)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(47), "-", 1)

'als er een > (=Chr(62)) in de naam voorkomt, vervangen door '

If InStr(1, sBedrijf, Chr(62)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(62), Chr(39), 1)

'als er een < (=Chr(60)) in de naam voorkomt, vervangen door '

If InStr(1, sBedrijf, Chr(60)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(60), Chr(39), 1)

'als er een * (=Chr(42)) in de naam voorkomt, vervangen door '

If InStr(1, sBedrijf, Chr(42)) > 0 Then sBedrijf = Replace(sBedrijf,
```

```
Chr(42), Chr(39), 1)
```

```
'als er een ? (=Chr(63)) in de naam voorkomt, vervangen door ')
```

```
If InStr(1, sBedrijf, Chr(63)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(63), Chr(39), 1)
```

```
'als er een | (=Chr(124)) in de naam voorkomt, vervangen door ')
```

```
If InStr(1, sBedrijf, Chr(124)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(124), Chr(39), 1)
```

```
'als laatste de eventuele dubbele '' vervangen door ')
```

```
sBedrijf = Replace(sBedrijf, Chr(39) & Chr(39), Chr(39), 1)
```

```
'als er een : (=Chr(58)) in de naam voorkomt, vervangen door '.')
```

```
If InStr(1, sBedrijf, Chr(58)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(58), ".", 1)
```

```
'als er een ; (=Chr(59)) in de naam voorkomt, vervangen door '.')
```

```
If InStr(1, sBedrijf, Chr(59)) > 0 Then sBedrijf = Replace(sBedrijf,
Chr(59), ".", 1)
```

```
'en de eventuele dubbele .. vervangen door .)
```

```
sBedrijf = Replace(sBedrijf, "..", ".", 1)
```

```
Return
```

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

```
TijdLoop:
```

```
'Om te voorkomen dat de programmatuur zich verslikt,
```

```
een kleine pauze van 1 seconde...
```

```
Start = 0
```

```
Start = Timer ' Aanvangstijd instellen.
```

```
Do While Timer < Start + 1
```

```
DoEvents ' Overdragen aan andere processen.
```

```
Loop
```

```
Return
```

De subroutine bevat aardig wat regels, maar er zijn ook nogal wat tekens die niet mogen in een bestandsnaam. En als je een bestandsnaam automatisch wilt genereren, dan wil je natuurlijk voorkomen dat de procedure vastloopt omdat er een verkeerde bestandsnaam tussen zit.

De laatste subroutine (TijdLoop) kun je gebruiken om een pauze in te bouwen, als de procedure bijvoorbeeld de handelingen te snel achter elkaar uitvoert waardoor een volgende stap niet goed kan worden gestart. Denk bijvoorbeeld aan het exporteren van het Excel bestand, terwijl tegelijkertijd Word al wordt geopend. Je kunt geen bestand vervangen als dat al in een ander pakket is geopend, dus om te voorkomen dat Word te vroeg opstart, zorg je er voor dat het opslaan van het Excel bestand voldoende tijd krijgt.

Samenvatting

In deze oefening worden verschillende Office bestanden gebruikt om een complex proces te automatiseren. Hierbij wordt vanuit Access een mailmerge in Word uitgevoerd, zonder dat de gebruikers zelf Word hoeft te openen. Andersom kan uiteraard ook: je kunt ook vanuit Word of Excel een automatisering in Access uitvoeren. Maar dat is voor een volgend hoofdstuk.