



# VBA voor doe het Zelfers - deel 10

Handleiding van Helpmij.nl

Auteur: leofact

Oktober 2014

“ Dé grootste en gratis computerhelpdesk van Nederland ”

VBA voor Doe het Zelfers is een reeks artikelen, bedoelt voor mensen die met VBA in Excel aan de slag willen om taken te automatiseren of om deze toegankelijk te maken voor gebruikers met weinig Excel kennis. VBA is een volwaardige programmeertaal met een woordenschat en een zinsopbouw (syntaxis). Om deze goed te kunnen leren wordt het aangeraden om hierover boeken te lezen. Bijvoorbeeld uit deze [willekeurige selectie](#).

## Vorige aflevering

De vorige keer hebben we de foutafhandeling in de schijnwerpers gezet. Dat was misschien niet het meest sexy onderwerp, maar noodzakelijk wanneer je een toepassing wilt maken die stabiel, gebruiksvriendelijk en zonder onprettige verassingens werkt.

## In deze aflevering

Dit keer gaan we Excel misbruiken om op verschillende manieren geautomatiseerd e-mail te versturen. Dit met toegevoegde inhoud en met én zonder bijlages. Ook dit keer is er een bijlage met alle uitgewerkte procedures welke je [hier](#) kunt downloaden.

## E-mail versturen via VBA

Het is met behulp van VBA mogelijk om op verschillende manieren een e-mail te genereren. Dat kan handig zijn binnen een bepaald werkproces of binnen een bepaalde workflow. Er zijn verschillende methodes om een e-mail te versturen. Drie hiervan worden in dit artikel behandeld.

## Hyperlink

De eerste manier is om de **FollowHyperlink** methode te gebruiken. Er wordt dan van de mailto functie gebruikt gemaakt. Het voordeel is dat deze methode op vrijwel alle systemen en voor bijna alle e-mail clients zal werken. Daar staat als nadeel tegenover dat de functie beperkte mogelijkheden biedt. Daarnaast wordt de e-mail niet direct verstuurd. Er wordt alleen een e-mail geopend. De gebruiker moet zelf nog op verzenden klikken. Dat geeft dan wel de mogelijkheid om de e-mail qua inhoud aan te vullen. Het hoeft dus geen bezwaar te zijn dat de e-mail handmatig verzonden moet worden. De Hyperlinkfunctie is bijvoorbeeld goed te gebruiken in een foutafhandelings-routine. Je kunt daarmee de gebruiker in de gelegenheid stellen de opgetreden fout te melden (en misschien zijn of haar frustratie daarover af te reageren). Een voorbeeld met gebruik van Hyperlink:

```

(Algemeen) Hyperlink_Mail
'-----
' Procedure : Hyperlink_Mail 23-8-2014
' Doel : mail versturen m.b.v. de methode hyperlink
'-----
'
Sub Hyperlink_Mail()
Dim sOntvanger As String
Dim sOnderwerp As String
Dim sLink As String
Dim sBody As String

'geef de variabelen de correcte mailinhoud
sOntvanger = "leofact1@gmail.com"
sOnderwerp = "Dit is een test met hyperlink"
sBody = "Goedendag" & Chr(11) _
        & "Dit is een voorbeeld mail." & Chr(11) _
        & "Gegeneerd door VBA." & Chr(11) _
        & "Vanuit de bijlage van de Helpmij" _
        & " nieuwsbrief van september 2014" & Chr(11) _
'maak een string, geschikt voor de mailo functie:
sOntvanger = "mailto:" & sOntvanger
sOnderwerp = "?subject=" & sOnderwerp
sBody = "&body=" & sBody
sLink = sOntvanger & sOnderwerp & sBody
ActiveWorkbook.FollowHyperlink (sLink)

End Sub

```

Het grootste deel van de hier getoonde code bestaat uit het vullen van de tekst variabelen. De eigenlijke code bestaat slechts uit één regel (de laatste).

## Foutafhandeling

De Hyperlink methode zou op de volgende manier in een foutafhandeling-routine kunnen worden opgenomen. Als eerste de mailfunctie zelf:

```

(Algemeen) Foutmelding_Mail
'-----
' Procedure : Foutmelding_Mail 24-8-2014
' Doel : Verzamelen van de foutmeldingen en door mailen.
'-----
'
Sub Foutmelding_Mail(ByVal sOnderwerp As String, _
                    ByVal sProcNaam As String, ByVal sFout As String)

Dim sOntvanger As String
Dim sBody As String
Dim sLink As String

sOntvanger = "mailto:" & "leofact@gmail.com"
sOnderwerp = "?subject=" & sOnderwerp
sBody = "&body=" & sBody
sLink = sOntvanger & sOnderwerp & sBody
ActiveWorkbook.FollowHyperlink (sLink)

End Sub

```

De routine verzamelt de inhoud van de foutmelding en zet de e-mail klaar. Om de fout zelf af te handelen is dan de volgende foutafhandeling-routine nodig:

```

(Algemeen) | Fout_Demo
'-----
' Procedure : Fout_Demo 24-8-2014
' Doel : Demonstreer een foutmelding
'-----
'
Sub Fout_Demo ()
Dim A As Long

    On Error GoTo Fout_Demo_Error

'Dit wordt vervangen door de eigen code:
    A = 1 / 0

    On Error GoTo 0
Exit Sub
'-----
' Procedure : Error handler
'-----
Fout_Demo_Error:
Dim sOnderwerp As String
Dim sFout As String
Dim sProcNaam As String

    sOnderwerp = "Foutmelding: " & Err.Number
    sProcNaam = "Fout_Demo"
    sFout = Err.Description
    Foutmelding_Mail sOnderwerp, sFout, sProcNaam
    Application.EnableEvents = True
End Sub
    
```

Het gaat hier natuurlijk om het deel van de Error handler. Eerder hebben we al MZ-Tools gebruikt (zie [het vorige artikel](#)). Wanneer deze code in de Error handler template van de Tool wordt gezet is het vervolgens appeltje-eitje om alle procedures van de foutmeldingsroutine te voorzien. Dit zou er zo uit kunnen zien in het betreffende venster van MZ-Tools:

```

Options
General | Nomenclature | Error Handler | Procedure Header | Module Header | Shortcuts | Code Templates | Toolbar | External Utilities
Variables: Procedure Body [Add]
On Error GoTo {PROCEDURE_NAME}_Error
    {PROCEDURE_BODY}
On Error GoTo 0
Exit {PROCEDURE_TYPE}
'-----
' Procedure : Error handler
'-----
{PROCEDURE_NAME}_Error:
Dim sOnderwerp As String
Dim sFout As String
Dim sProcNaam As String
Application.EnableEvents = True
    sOnderwerp = "Foutmelding: " & Err.Number
    sProcNaam = {PROCEDURE_NAME}
    sFout = Err.Description
    Foutmelding_Mail sOnderwerp, sFout, sProcNaam
    
```

Donate OK Cancel Help

## Sendmail

Een andere methode om e-mail te verzenden is met behulp van **Sendmail**. Het is heel gemakkelijk om een e-mail te versturen met behulp van deze methode. Met het volgende regeltje verstuurt u een e-mail met het actieve werkboek als bijlage.

```
ActiveWorkbook.SendMail "leofact1@gmail.com", "testmail"
```

Ontvanger en het onderwerp zijn hierin ook al gespecificeerd. De methode heeft echter behoorlijke beperkingen. CC en BCC zijn niet mogelijk. Ook het toevoegen van body tekst is niet mogelijk. Bovendien wordt de mail pas verzonden als het email programma geopend wordt (het staat tot die tijd in postvak uit).

Voordat het daarin komt, triggert het een beveiliging bij Outlook. Deze last een pauzeperiode in en pas daarna kun je de e-mail toevoegen. Deze beveiliging kan overigens ook door één van de andere mail methoden worden getriggerd, maar vanwege de andere beperkingen van Sendmail laat ik het verder hierbij wat Sendmail betreft.



## Outlook Object

Een goede manier om e-mail te verzenden is via het Outlook Object. Daarbij spreekt het belangrijkste nadeel al voor zichzelf; Outlook moet namelijk op het systeem geïnstalleerd zijn. Gelukkig is dit in veel bedrijfsmatige omgevingen het geval. De toegang tot Outlook wordt verkregen met de CreateObject methode, zoals in dit voorbeeld:

```

(Algemeen) OutlookMailDemoTekst
'-----
' Procedure : OutlookMailDemoTekst 23-8-2014
' Doel : Mail verzenden met een tekst body
'-----
'
Sub OutlookMailDemoTekst()
Dim sBody As String

    On Error GoTo OutlookMailDemoTekst_Error

    sBody = "Goedendag," & vbNewLine & vbNewLine & _
           "Dit is de eerste regel" & vbNewLine & _
           "Dit is de tweede regel" & vbNewLine & _
           "Dit is de derde regel" & vbNewLine & _
           "En dit de laatste regel."

    With CreateObject("Outlook.Application").CreateItem(olMailItem)
        .To = "leofact1@gmail.com"
        .CC = ""
        .BCC = ""
        .Subject = "Onderwerp regel TEST"
        .Body = sBody
        .Send 'display
    End With

    On Error GoTo 0
    Exit Sub

'-----
'Foutafhandeling
'-----
OutlookMailDemoTekst_Error:
Application.EnableEvents = True
MsgBox "Error " & Err.Number & " (" & Err.Description & _
      ") in procedure OutlookMailDemoTekst of Module Module1"

End Sub

```

Omdat

Outlook ook een VBA bibliotheek kent kan er gebruik gemaakt worden van VBA constanten. vbNewline zorgt hier voor een regelafbreking.

Het is goed mogelijk om e-mail naar meerdere geadresseerden te sturen. Men kan bijvoorbeeld met een klein lusje de e-mail adressen uit een kolom lezen en aan één van de adres velden toevoegen. De uitwerking hiervan is goed zelf te doen.

Met Send wordt de e-mail direct verzonden. Outlook wordt netjes geopend op de achtergrond, het mailtje wordt verzonden en Outlook wordt weer keurig gesloten. Helaas kan ook hier kan de beveiligings- "teaser" getriggerd worden. In dat geval is het niet mogelijk om meerdere e-mails achter elkaar te versturen en wordt de wachttijd waarschijnlijk als zeer irritant ervaren. Helaas is mij geen manier bekend waarop deze beveiliging kan worden uitgeschakeld. Het kan dan een alternatief zijn om Display te gebruiken in plaats van Send. De mail wordt nu alleen klaar gezet. Er moet dan door de gebruiker nog op verzenden worden geklikt. Dat biedt gelijk de mogelijkheid om nog aanpassingen te maken in de mail voor deze wordt verzonden. Verzenden kan dan eventueel met de sneltoets CTRL + Enter. Dit moet echter wel eenmalig zijn ingesteld door de gebruiker. Het gebruik van de sneltoets is natuurlijk te simuleren met de methode SendKeys en het is dan ook verleidelijk om dit te proberen. Dit zou als volgt kunnen:

```

(Algemeen) OutlookMailDemoSendKeys
' Procedure : OutlookMailDemoSendKeys 23-8-2014
' Doel : Mail verzenden met het actieve werkboek als bijlage
'-----
'
Sub OutlookMailDemoSendKeys()
Dim sBody As String

    sBody = "Goedendag" & Chr(11)
        & "Dit is een voorbeeld mail." & Chr(11)
        & "Gegeneerd door VBA." & Chr(11)
        & "Vanuit de bijlage van de Helpmij"
        & " nieuwsbrief van september 2014"

    With CreateObject("Outlook.Application").CreateItem(olMailItem)
        .To = "leofact1@gmail.com"
        .CC = ""
        .BCC = ""
        .Subject = "Bijlage TEST JA!"
        .Body = sBody
        .display
    End With
    'Sneltoetsen: CTRL + Enter
    Application.SendKeys ("^~")
End Sub

```

Deze methode is echter zo weinig robuust dat het gebruik ervan waarschijnlijk meer problemen zal opleveren dan dat er gemak van is te verwachten.

## Bijlage

Net als bij Sendmail kan ook met de Outlook Object methode het actieve werkboek als bijlage worden toegevoegd. Het werkboek dient echter wel opslagen te zijn en met pad en bestandsnaam te worden toegevoegd als volgt:

```

(Algemeen) OutlookMailDemoBijlage
End Sub
'-----
' Procedure : OutlookMailDemoBijlage 23-8-2014
' Doel : Mail verzenden met het actieve werkboek als bijlage
'-----
Sub OutlookMailDemoBijlage()
Dim sBody As String

On Error GoTo OutlookMailDemoBijlage_Error

sBody = "Goedendag" & Chr(11) _
        & "Dit is een voorbeeld mail." & Chr(11) _
        & "Gegenereerd door VBA." & Chr(11) _
        & "Vanuit de bijlage van de Helpmij" _
        & " nieuwsbrief van september 2014"

'sla de laatste wijzingen op
ActiveWorkbook.Save
'maak de mail.
With CreateObject("Outlook.Application").CreateItem(olMailItem)
.attachments.Add ActiveWorkbook.FullName
.To = "leofact1@gmail.com"
.CC = ""
.BCC = ""
.Subject = "Bijlage TEST"
.Body = sBody
.Send 'display
End With

On Error GoTo 0
Exit Sub

'-----
OutlookMailDemoBijlage_Error:
Application.EnableEvents = True
Call MsgBox("Excel kan gaan mail aanmaken." _
            & vbCrLf & "Outlook is niet toegankelijk of afwezig." _
            , vbCritical, "Mail fout")
End Sub

```

Foutafhandeling is hierbij aan te bevelen, omdat de methode natuurlijk alleen op systemen werkt waarop Outlook is geïnstalleerd.

## Werkblad als bijlage

Een enkel werkblad als bijlage mee sturen kan niet zonder meer. Daarvoor zal het eerst gekopieerd moeten worden naar een nieuw werkboek. Dit moet vervolgens worden opgeslagen als tijdelijk bestand en met de bestandsnaam als bijlage worden toegevoegd. Daarna dient het tijdelijke bestand verwijderd te worden.

Deze laatste stap kan natuurlijk desgewenst worden overgeslagen. De volgende routine doet dit alles door een tijdelijk werkboek aan te maken in dezelfde map als waarin het in gebruik zijnde werkboek is opgeslagen (dat moet dus wel zijn opgeslagen).



```

(Algemeen) OutlookMailBijlage
'-----
' Procedure : OutlookMailBijlage 31-8-2014
' Doel : mail een werkblad als bijlage
'-----
'
Sub OutlookMailBijlage()
Dim sBody As String
Dim sPad As String
'pad van de bijlage is hetzelfde als van het werkboek
sNaam = ActiveWorkbook.Path & "/" & "Factuur DATA.xls"
'alle Excel berichten uit.
Application.DisplayAlerts = False
'kopieer het werkblad naar een nieuwe werkboek.
Blad1.Copy
'sla dit op en sluit het.
With ActiveWorkbook
    .SaveAs sNaam
    .Close False
End With
'maak de mail
sBody = "Goedendag" & Chr(11) _
        & "Dit is een mail met een werkblad als bijlage."

With CreateObject("Outlook.Application").CreateItem(olMailItem)
    .To = "leofact1@gmail.com"
    .CC = ""
    .BCC = ""
    .Subject = "Onderwerp regel TEST"
    .Body = sBody
    'voeg het tijdelijke werkboek toe als bijlage.
    .attachments.Add sNaam
    .Send 'display
End With
'verwijder de bijlage
Kill sNaam

End Sub

```

## Nog meer voorbeelden

Er zijn nog vele voorbeelden denkbaar waarbij de inhoud van een range, of één of meerdere werkbladen als body worden toegevoegd aan de e-mail. Deze voorbeelden zijn op internet te vinden. Twee sites daarvan noem ik speciaal, waarvan er één vele duidelijk voorbeelden toont en ook nog in het Nederlands is. Dat is de al eerder genoemde [www.snb-vba.eu/alfabet\\_en.html](http://www.snb-vba.eu/alfabet_en.html). De andere is [www.rondebruin.nl/win/s1/outlook/mail.htm](http://www.rondebruin.nl/win/s1/outlook/mail.htm)

Met Google zijn er nog vele voorbeelden te vinden.

## HTML

Een laatste voorbeeld waar hier nog op ingegaan wordt is de mogelijkheid om HTML opmaak aan de body toe te voegen. Daarmee kan bijvoorbeeld een tabel worden gegenereerd waar een lijst gegevens overzichtelijk in gepresenteerd kunnen worden. In het voorbeeld wordt een lijst facturen weergegeven. Deze kunnen zijn betaald, nog open staan, of over tijd zijn. De routine maakt het mogelijk om de lijst aan een e-mail toe te voegen. Eerst wordt de data naar een apart werkblad gekopieerd en dan pas opgemaakt en de HTML toegevoegd. Dit houdt data en opmaak gescheiden, wat meer controle over de correcte inhoud van data geeft. Nadeel is wel dat de routine langer wordt. De routine wordt (in verband met de lengte) hier gegeven in twee delen. Het eerste deel zet de data klaar en zorgt voor de opmaak. In het tweede deel wordt de e-mail daadwerkelijk klaar gezet of verzonden. Ten slotte worden er nog twee kleine functies en een subroutine gebruikt om de opmaak iets efficiënter te laten lopen. Uitleg wordt gegeven door middel van de commentaar tekst. Deel 1:

```

(Algemeen) MailTabel
'-----
' Procedure : MailTabel 30-8-2014
' Doel : Maak een tabel van de gegevens en zet deze in de mail
'-----
Sub MailTabel()
Dim rij As Long
Dim lRij As Long
Dim lX As Long
Dim lY As Long
Dim sHTML As String
Dim sTabel As String
Dim sDatum As String
Dim sAdres As String
Dim sFont As String
Dim Sh As Worksheet
Set Sh = Blad1

On Error GoTo MailTabel_Error
With Application
.ScreenUpdating = False
.EnableEvents = False
End With
sDatum = Strings.Format(Now, "dd-mm-yyyy hh:mm")
sAdres = "Leofact1@gmail.com"
sOnderwerp = "Factuur overzicht dd." & sDatum

With Blad9
'maak de range leeg
.Range("A15:F50").ClearContents
'berichtkop
.Cells(1, 1) = "Facturen overzicht dd.: " & sDatum
'naamvelden rij
.Cells(2, 1) = Spatie("naam")
.Cells(2, 2) = Spatie("factuurnr.")
.Cells(2, 3) = Spatie("factuurdatum")
.Cells(2, 4) = Spatie("betaald")
'kop te laat en niet betaald.
.Cells(3, 1) = sB & "Niet betaald, te laat:"
rij = 3
'te laat en niet betaald.
.Cells(rij, 1) = Spatie("Niet betaald, te laat:")
For lRij = 15 To 40
If Left(Sh.Cells(lRij, 6), 1) = "t" Then
rij = rij + 1
'roep de sub aan die de rij vult.
CelRij rij, lRij
End If
Next lRij
rij = rij + 1
'niet betaald, nog niet te laat.
.Cells(rij, 1) = sB & "Niet betaald, nog niet te laat:"
For lRij = 16 To 40
If Left(Sh.Cells(lRij, 6), 1) = "n" Then
rij = rij + 1
CelRij rij, lRij
End If
Next lRij
rij = rij + 1
'op tijd betaald.
.Cells(rij, 1) = sB & "Betaald en afgerond:"
For lRij = 15 To 40
If Left(Sh.Cells(lRij, 6), 1) = "j" Then
rij = rij + 1
CelRij rij, lRij
End If
Next lRij
rij = rij + 1
'Start met de HTML.+++++
'Maak de container zé de header klaar
sHTML = "<body><basefont color=""gray""/>" _
& "<h3>" & .Cells(1, 1) & "</h3>" _
& "<table border=1 bgcolor=FAFAFB>"
'tabeluitlijning
sTabel = "<TD VALIGN=""top"">"
'loop alle rijen langs
For iY = 3 To rij - 1
'maak een tablerow
sHTML = sHTML & "<tr>"
'combineer cellen als de inhoud een kop is.
If .Cells(iY, 5) = "" Then
sHTML = sHTML & "<TD bgcolor = #F6F7DE COLSPAN=""4"" >" _
& "<font color=#084B8A>" & .Cells(iY, 1) & "</td></tr>"
Else
'geen kop, vul dan de vier cellen.
For iX = 1 To 4
'reguliere tekst zwart.
sFont = "><font color=""black"">"
'te laat? Dan rode tekst.
If Left(.Cells(iY, 5), 1) = "t" Then
sFont = "><font color=""red"">"
'veldnamen rij? Dan aangepaste kleur.
ElseIf InStr(1, .Cells(iY, 1), "naam") = 1 Then
sFont = "><font color=#A2B4C3>"
End If
'voeg toe aan de HTML string
sHTML = sHTML & sTabel & sFont & .Cells(iY, iX) & "</td>"
'volgende cel
Next iX
'sluit de rij met cellem af.
sHTML = sHTML & "</tr>"
End If
'volgende rij
Next iY
End With
'sluit de container
sHTML = sHTML & "</table></body>"

```

Deel twee om de mail te verzenden op de inmiddels bekende wijze:

```
'open de mail en plaats de inhoud. ++++++
With CreateObject("outlook.application").CreateItem(0)
    .To = sAdres
    .Subject = sOnderwerp
    .htmlbody = sHTML
    .display
End With
Application.EnableEvents = True

On Error GoTo 0
Exit Sub

'-----
' Procedure : Error handler
'-----
MailLabel_Error:
Call MsgBox("De bespreeklijst mail kan niet worden verzonden." _
    & vbCrLf & "Mogelijk is MS Outlook niet aanwezig, of niet toeg
    & vbCrLf & "" _
    , vbCritical, "Mail fout")
Application.EnableEvents = True

End Sub
```

En vervolgens de opmaak functies:

```
(Algemeen) Spatie
Function Datum(dDatum As Date) As String
'Geef de datum het juiste format
Datum = Format(dDatum, "dd-mm-yyyy")
End Function
Function Euro(lEuro As Long) As String
'Geef het bedrag het juiste format
Euro = Format(lEuro, "currency")
End Function
Function Spatie(sSpatie As String) As String
'Maak met een HTML spatie ruimte vanaf de borders.
Spatie = "&nbsp;" & sSpatie & "&nbsp;"
End Function
```

Voel je vooral uitgenodigd om met deze routine, de bijbehorende gegevens en de opmaak in de bijlage te gaan spelen.

## SMTP

Het is ook mogelijk om e-mail direct via het mailsysteem (SMTP) van de computer te verzenden. Dit kan met behulp van CDO en heeft als voordeel dat mail onafhankelijk van het geïnstalleerde mailprogramma kan worden verzonden. Er is echter kennis van het e-mailsysteem vereist en het wordt hier niet verder behandeld. Voorbeelden zijn o.a. hier te vinden:

[www.rondebruin.nl/win/s1/cdo.htm](http://www.rondebruin.nl/win/s1/cdo.htm)

Hiermee besluit ik deze aflevering.

## Samenvatting:

Er zijn vele manieren en mogelijkheden om e-mail te verzenden met behulp van VBA, de methodes Hyperlink mailto, Sendmail en Outlook Object werden behandeld met voorbeelden daarvan, met en zonder bijlage en met en zonder inhoud tekst. Daarbij werden de voor en nadelen van deze methodes besproken met verschillende manieren om daar mee om te gaan. Er werd ook een methode gegeven om meer controle op de opmaak uit te oefenen met behulp van HTML. Als laatste werd de mogelijkheid om mail te verzenden via de SMTP genoemd. Dit werd niet verder uitgewerkt.

## Volgende aflevering

De volgende aflevering ga ik in op het werken met datums en het rekenen daarmee. Daarnaast wordt er een manier gegeven om datums in te voeren met behulp van de Date & Time Picker Control 6.0.