



# VBA voor Doe het Zelfers deel 6

Handleiding van Helpmij.nl

Auteur: leofact

Mei 2014

“ Dé grootste en gratis computerhelpdesk van Nederland ”

## Vorige aflevering

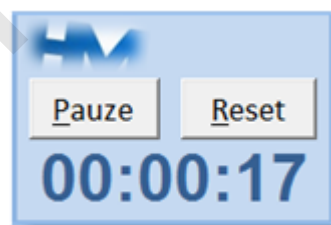
De vorige keer werd de Event of gebeurtenis macro besproken met verschillende voorbeelden die ook in de bijlage werden uitgewerkt. Verder werd een manier geboden om ongewenst knippen en plakken tegen te gaan en werd er een simpele klok gemaakt.

## In deze aflevering

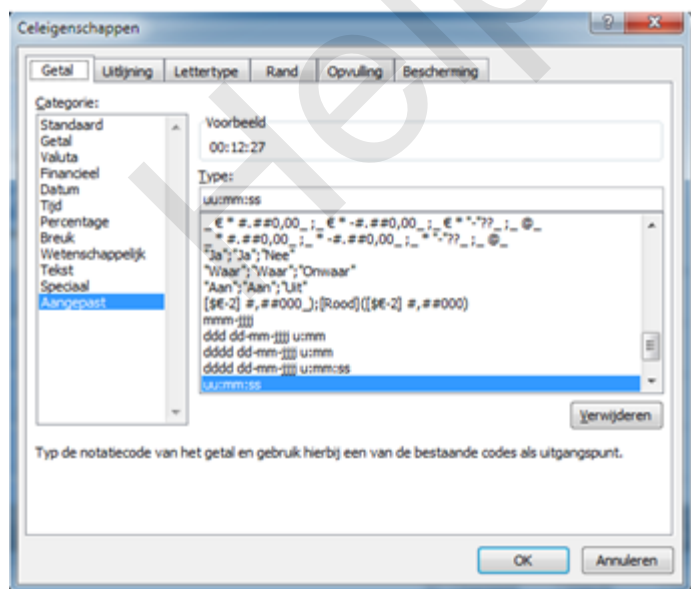
Dit keer wordt er een timer gecreëerd met behulp van de OnTime methode. Er wordt ook een methode gegeven om, met behulp van een klein beetje VBA en voorwaardelijke opmaak, de geselecteerde rij en kolom zichtbaar te maken. Daarnaast wordt de conversie van Excel 2003 naar 2010 besproken en worden een aantal van de problemen die daarbij op kunnen treden behandeld. Ook bij deze aflevering hoort een bijlage die [hier](#) kan worden gedownload.

## De timer

De timer die we gaan bouwen wordt een soort stopwatch, met een start-, pauze- en een resetfunctie. In de vorige aflevering is al besproken dat de OnTime methode niet 100% nauwkeurig is en dat de kleinste te gebruiken eenheid de seconde is. Rekening houdend met deze beperkingen kan er toch een redelijke timer worden gecreëerd met deze methode. We starten met een cel aan te wijzen waarin de tijd gaat komen.



De cel-eigenschappen (CTRL + 1) van deze cel worden op “aangepast” ingesteld:



Stel vervolgens het lettertype naar wens in. Maak daarna twee knoppen uit de ActiveX objecten ([zie aflevering 4](#)) en geef deze de volgende namen: CbutTimer en CbutReset. Stel de knop tekst in m.b.v. Caption en maak daar heel origineel “Start” en “Reset” van. Maak eventueel met cel-opmaak de vormgeving af. Dit is voor de werking natuurlijk verder overbodig. Nu is alleen nog de code nodig om de timer te laten functioneren. Hiervoor is een routine nodig die de timer start.

Vervolgens één die deze laat lopen en een routine die de timer pauzeert. Als laatste is er een routine nodig om de timer te kunnen resetten. Om de benodigde waarden door te geven worden hiervoor drie globale variabelen gebruikt:

```
(Algemeen) Timer
Public dStarttime As Date
Public dVolgendeTijd As Date
Public TimerOn As Boolean
```

Uitleg over variabelen is in [deel 3](#) te vinden.

De volgende stap is de bouw van een routine waarin de timer aan de gang wordt gehouden en die de tijd in de aangewezen cel zet. In deze routine wordt in er in de eerste regel alvast rekening mee gehouden dat de Timer gestopt moet kunnen worden als TimerOn de waarde "False" heeft gekregen. Wanneer de waarde "True" is, wordt er uitgerekend wat de huidige tijd is van de timer. Hiervoor wordt de starttijd van de huidige tijd afgetrokken. Dit wordt vervolgens op de juiste plaats op het werkblad gezet met behulp van de Format Functie. Strikt genomen is Format hier niet nodig, daar de gewenste opmaak ook al in de cel-eigenschappen is aangegeven. Hierna wordt de volgende tijd uitgerekend door een seconde bij de huidige tijd op te tellen. Hiervoor wordt de functie TimeValue gebruikt. Als laatste wordt deze timer routine weer aangeroepen door de OnTime methode op de ingestelde tijd. Het geheel kan er dan zo uitzien:

```
(Algemeen) TimerStart
End Sub
'-----
' Procedure : Timer
' Purpose   : Laat de timer lopen
'-----
Sub Timer()
Dim dVerschil As Date
' zet de timer uit als deze moet stoppen.
If TimerOn = False Then
    Application.OnTime dVolgendeTijd, "Timer", Schedule = False
    Exit Sub
End If
' zet de timer een seconde verder
dVerschil = Now - dStarttime
Blad1.Cells(13, 5) = Format(dVerschil, "hh:mm:ss")
Application.EnableEvents = True
'tel een seconde bij de huidige tijd om timer in te stellen
dVolgendeTijd = Now + TimeValue("00:00:01")
'laat de timer lopen
Application.OnTime dVolgendeTijd, "Timer"
End Sub
```

Nu moet de Timer natuurlijk nog worden gestart. Hiervoor maken we een routine waarin gekeken wordt of de timer voor het eerst wordt gestart, of eerder werd gepauzeerd door de gebruiker. Als de timer voor het eerst wordt gestart zal de waarde van dStarttime 0 zijn. Daar wordt op gecontroleerd. Als dit het geval is wordt dStarttime op Now (nu) ingesteld. In het tweede geval wordt de verlopen tijd die de timer aangeeft daarvan afgetrokken. Op deze manier loopt de Timer weer verder waar deze gebleven was:

```

(Algemeen) Timer
'-----
' Procedure : TimerStart
' Purpose   : Zet de timer in gang
'-----
'
'
Sub TimerStart()
    'stel dStarttime in
    If dStarttime = 0 Then
        'als deze de eerste keer wordt gebruikt
        dStarttime = Now
    Else
        'of laat deze doorlopen
        dStarttime = Now - Blad1.Cells(13, 5)
    End If
    Timer
End Sub

```

Nu kun je op de startknop klikken wat je wilt, maar er gebeurt nog steeds niets. Daarvoor is het nodig dat de klik wordt afgevangen in het Click event. Daarin regelen we gelijk beide mogelijkheden van de knop. Bij de eerste klik zetten we de Caption (knoptekst) op "Pauze" en starten door TimerOn de waarde "True" te geven en vervolgens naar TimerStart te springen. In het andere geval geven we TimerOn de waarde "False" en zetten de Caption op "Start" om de gebruiker te informeren over de functie van knop:

```

CbutTimer Click
'-----
' Procedure : CbutTimer_Click
' Purpose   : Start of stop de timer
'-----
'
'
Private Sub CbutTimer_Click()

With Blad1.CbutTimer
    'Start de Timer
    If .Caption = "Start" Then
        .Caption = "Pauze"
        TimerOn = True
        TimerStart
    Else
        'Pauze
        TimerOn = False
        .Caption = "Start"
        .Accelerator = "P"
    End If
End With
End Sub

```

Nu kan de Timer daadwerkelijk gestart worden. Ons rest alleen nog een routine om de timer te resetten. Deze zouden we gelijk in het Click event van de Resetknop kunnen zetten. Het is echter netjes om de timer gereset achter te laten als het werkboek wordt gesloten. We maken daarom een aparte routine die ook vanuit het Workbook\_BeforeClose event kan worden opgeroepen. De reset routine kan er bijvoorbeeld zo uitzien:

```

(Algemeen) | GeenCopyPaste
'-----
' Procedure : TimerReset
' Purpose   : reset de timer
'-----
Sub TimerReset()
On Error Resume Next
'OnTime uitzetten
Application.OnTime dVolgendeTijd, "Timer", Schedule = False
TimerOn = False
dVolgendeTijd = 0
With Blad1
'Timer weer het oorspronkelijke uiterlijk geven
.CbutTimer.Caption = "Start"
.Cells(13, 5) = "00:00:00"
End With
End Sub
    
```

In het Timerbutton Click event hoeft dan alleen maar deze routine te worden aangeroepen:

```

CbutReset | Click
'-----
' Procedure : CbutReset_Click
' Purpose   : Reset Routine aanroepen
'-----
Private Sub CbutReset_Click()
TimerReset
End Sub
    
```

Ditzelfde kan nu ook in het BeforeClose event:

```

Workbook | BeforeClose
'-----
' Procedure : Workbook_BeforeClose
' Purpose   : Wijs de standaardfunctie weer toe aan de sneltoetsen
'-----
Private Sub Workbook_BeforeClose(Cancel As Boolean)
With Application
.OnKey "^c"
.OnKey "^x"
.OnKey "^v"
.OnKey "+{DEL}"
.OnKey "^{INSERT}"
End With
'stop de timer
TimerReset
End Sub
    
```

Hiermee is de Timer klaar en stappen we naar het volgende onderdeel.

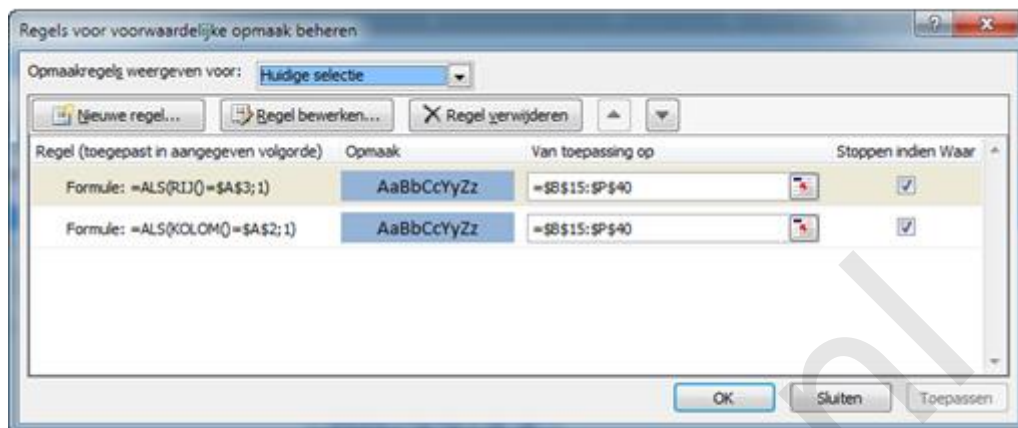
## Regel en rij selectie

In Excel is te zien in welke rij en kolom de geselecteerde cel zich bevindt. Soms wordt het belangrijk gevonden om dit duidelijker te accentueren. Dit kan er als volgt uit zien:

cat1	item1	Petersen, Willem	maandag 21 - 04 - 2014	post1	data1	data_a1
cat2	item2	Karelse, Piet	donderdag 22 - 05 - 2014	post2	data2	data_a2
cat3	item3	Hogendijk, Leo	woensdag 19 - 03 - 2014	post3	data3	data_a3

Dit kan heel eenvoudig bereikt worden door een combinatie van voorwaardelijke opmaak en een regeltje VBA. Nog niet bekend met voorwaardelijk opmaak? Lees dan eerst de uitleg hierover op [Office.Microsoft.com](http://Office.Microsoft.com)

Open het venster "regels beheren". Via de tab Start > Voorwaardelijk opmaak > Regels beheren. Maak daar de twee regels aan zoals hieronder getoond:



Geef een hierbij ook de gewenste opmaak aan. In de bijlage wordt de cel-opvulling op licht blauw gezet. Dit wordt getoond in die cellen waarvan de rij en kolomnummers overeen komen met de cellen A2 en A3. Dit wordt pas functioneel als daar steeds het nummer van de geselecteerde rij en kolom in staan. Hiervoor is een klein stukje VBA noodzakelijk. We kunnen dat realiseren met behulp van het SelectionChange event:



Dit is dan alles wat er nodig is. We gaan nu naar het laatste onderdeel van deze aflevering, namelijk:

## Werkboeken converteren

Microsoft brengt regelmatig nieuwe Office versies uit en dat is een mooi iets. Over het algemeen brengt dat namelijk nieuwe functies en verbeteringen met zich mee. Een grote overstap is de stap van Office 2003 naar Office 2007/2010. Dit vanwege de geheel andere gebruikerservaring door de introductie van het lint, maar ook door de ander manier van opslag en benaming van de bestanden. Werken met het lint is voor iedereen wennen. De één vindt het uiteindelijk zeer prettig en de ander blijft er mee worstelen.

Nu per 8 april de ondersteuning voor Office 2003 is beëindigd, wordt het écht tijd om over te stappen naar een nieuwere versie. De nieuwere Office versies maken gebruik van een ander bestandsformaat. Dit formaat heeft als voordeel dat de bestanden veel kleiner zijn. Meestal rond 50% kleiner dan de Office 2003 versies. Dit nieuwe formaat is ook nodig om de nieuwe functies te kunnen gebruiken. Er zijn ook nadelen: zo is het om beveiligingsredenen noodzakelijk om een werkboek met macro's in een ander bestandsformaat op te slaan. Namelijk in .xlsm of .xlsb in plaats van .xlsx. Doe je dit fout, dan zijn in één klap alle macro's verdwenen. Daar wordt gelukkig dan wel eerst voor gewaarschuwd.

Lastiger is dat een ouder werkboek wel eens corrupt wil raken en dan opent met de melding "Dit werkboek bevat onleesbare inhoud". Excel biedt dan wel een mogelijkheid tot herstel, maar die blijkt dan niet altijd opgeslagen te kunnen worden. Voor deze "noodsituatie" geef ik een manier om toch verder te kunnen en nog wat tips om de fout te voorkomen.

## Onleesbare inhoud

Als het een werkboek betreft dat oorspronkelijk opgezet werd in Excel 2003 kan het helpen dit weer terug te converteren. Kies wanneer de melding over de onleesbare inhoud verschijnt voor herstellen. Sla vervolgens met "Opslaan als" het werkboek op als Excel 2003 .xls. In veel gevallen kan het nu weer gewoon worden gebruikt.

## Converteren

Op de juiste manier converteren kan al een hoop narigheid schelen.

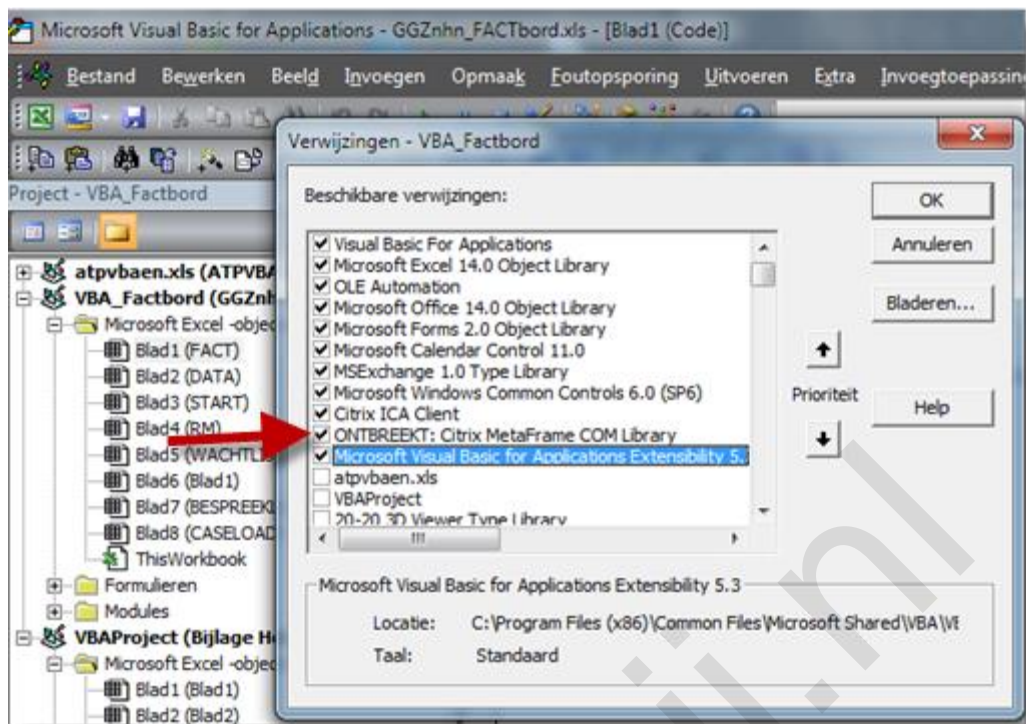
Met "Opslaan als" zijn .xls werkboeken op te slaan als .xlsx en .xlsm. Daarbij wordt het werkboek echter niet geconverteerd naar het nieuwe formaat. Doe dit door voor converteren te kiezen via Bestand > Info > Converteren.



Dat er nu anders wordt opgeslagen is gelijk merkbaar aan de veel kleinere bestandsgrootte die overblijft.

Bij werkboeken met macro's kan er ook na converteren een probleem ontstaan. Dit wordt mogelijk veroorzaakt door de beveiliging van Windows 7. Het advies was daarom ook om dit converteren op een XP PC te doen. Dat is nu een verouderd advies, daar XP niet meer ondersteund wordt. Wat dan wel kan helpen is om tijdens het converteren UAC (User Account Control) uit te zetten. Lees [hier](#) hoe dat gaat, wanneer je daar niet bekend mee bent.

Buiten deze maatregelen zijn er nog een aantal andere zaken die kunnen helpen. Zorg ervoor dat er geen ontbrekende referenties aangevinkt staan. Controleer dit door in de VBA Editor via menu Extra naar verwijzingen te gaan. Haal in ieder geval de vinkjes weg bij verwijzingen waar 'ontbreekt' bij staat.



In principe zijn in de meeste gevallen alleen de bovenste 5 verwijzingen noodzakelijk, maar dit kan per situatie verschillen. Hetzelfde kan gelden voor invoegtoepassingen (menu invoegtoepassingen > beheer van invoegtoepassingen). Verwijder daar alle niet noodzakelijke invoegtoepassingen. Het resultaat is het beste wanneer deze handelingen nog voor het converteren worden verricht.

Andere mogelijke oorzaken voor problemen na het converteren:

Een enkele keer kan een sorteerroutine corrupt raken. Voornamelijk als deze in de klasse module van een werkblad staan (bijvoorbeeld in de Worksheet\_Activate). Er wordt dan gemeld dat er een probleem is met een bepaald record op een bepaald werkblad. Opnieuw schrijven van de routine voor dat blad kan het probleem dan blijvend verhelpen.

Verder kan een probleem ontstaan met voorwaardelijk opmaak. Dit omdat de mogelijkheden hiervan veel uitgebreider zijn in de nieuwste office versies. De regels controleren en zo nodig aanpassen kan helpen. Dit laatste kan ook gelden voor draaitabellen.

Hiermee besluit ik deze aflevering.

Samenvattend:

In deze aflevering werd de OnTime methode verder besproken en met behulp van deze methode werd een Stopwatch gebouwd. Verder werd er een manier gegeven om de geselecteerde rij en kolom meer zichtbaar te maken. Dit werd bereikt met een combinatie van voorwaardelijke opmaak en wat VBA code. Als laatste werd de conversie van Office 2003 naar nieuwere versie besproken en werden er tips gegeven om dit beter te laten verlopen.

## Volgende aflevering

De volgende keer wordt het maken van keuzes besproken (o.a. If, Then, Else) en de (on)mogelijkheden van het nemen van sprongen door de routines.