



VBA voor Doe het Zelfers deel 12

Handleiding van Helpmij.nl

Auteur: leofact

November 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

In de vorige aflevering zijn we aan de slag gegaan met datums. Het format van de datum zetten we naar onze hand en we hebben er mee gerekend. Daarnaast werd er een handige manier van datum invoer getoond met behulp van de Microsoft Date and Time Picker 6.0.

In deze aflevering

Dit keer komen er verschillende manieren om fouten op te sporen aan bod, het zogenaamde debuggen. Daarbij worden ook mogelijkheden getoond om de inhoud van de gebruikte variabelen te tonen. Voorkomen is natuurlijk beter dan genezen. Daarom eerst een paar tips om een procedure meer gestructureerd en met minder kans op fouten op te zetten. Er is ook dit keer weer een bijlage beschikbaar. Dit werkboek is echter een bijlage van een eerdere aflevering en puur bedoelt om wat met de getoonde opties te experimenteren. Het heeft uiteraard de voorkeur wanneer dit toegepast wordt op een zelfgeschreven procedure.

Vooraf

Bij het opzetten van een project is het een goed idee om te beginnen met uit te denken wat het project precies gaat inhouden. Wat is het doel, of wat zijn de doelen? Met welke input wil je werken en wat moet er dan als output uitkomen? Welke variabelen heb je daarbij nodig en wat moet hun scope of levensduur zijn? Heb je meerdere procedures nodig? Hoe gaan die op elkaar aansluiten? Voor een beetje project kan dat al best een heel plan zijn. Stap voor stap de procedures en functies uitwerken aan de hand van een logisch uitgedachte volgorde kan dan gestructureerd werken bevorderen. Om het overzicht te bewaren kan het erg helpend zijn om een stroomschema of flowchart te maken. Dat kun je op verschillende manieren doen; bijvoorbeeld gratis en online bij [draw.io](#).

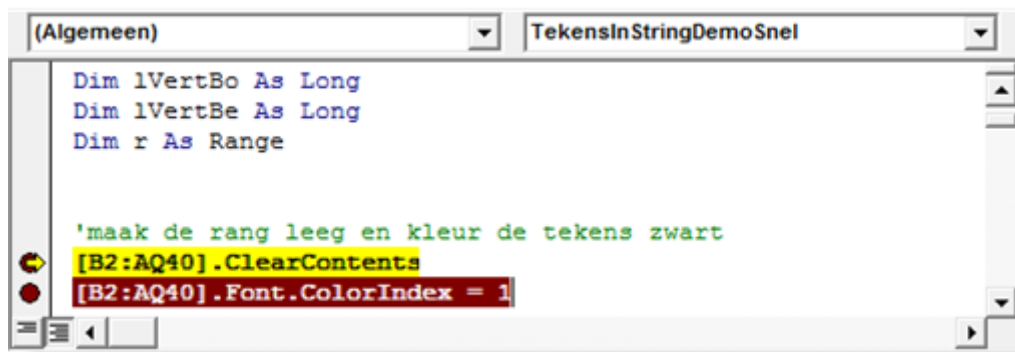
Hoe goed je ook programmeert, vroeg of laat gaat het een keer fout. De VBA Editor biedt je dan een aantal hulpmiddelen die hier worden besproken.

Stap voor stap

Voer de code stap voor stap uit door in de juiste procedure op F8 te drukken. Met Shift + F8 sla je een stap over en met CTRL + F8 wordt de code tot aan de cursor uitgevoerd. Handig om een bepaald blok code uit te laten voeren. De regel welke uitgevoerd wordt herken je doordat de achtergrond geel gekleurd is.

Onderbrekingspunt

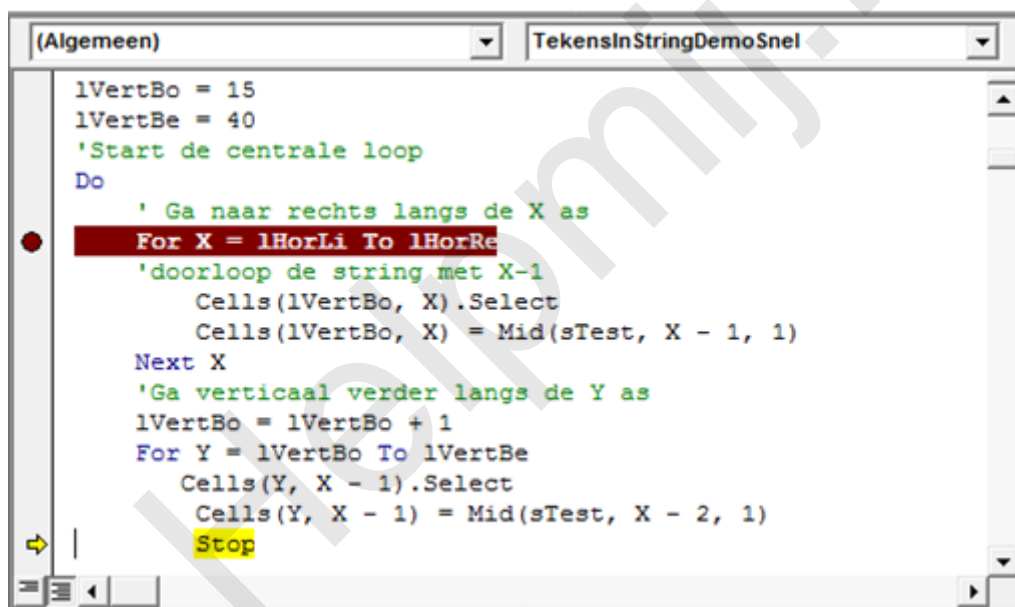
Met een onderbrekingspunt laat je de code uitvoeren tot het punt wat je hebt gemarkeerd. Dat kan door een klik in de linker zijbalk naast de code of door op F9 te drukken. Je kunt meerdere onderbrekingspunten plaatsen. Onderbrekingspunten herken je aan de donkerrode stip in de linker zijbalk. De tekst van de regel is in wit op een donkerrode achtergrond.



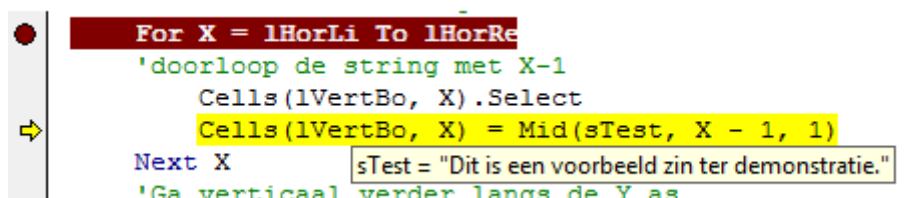
Onderbrekingspunten worden gewist door er weer op te klikken. Met CTRL + Shift + F9 worden ze allemaal in één keer gewist. Dat gebeurt automatisch wanneer het werkboek wordt afgesloten.

Stop

De programma uitvoering is ook te onderbreken met het commando Stop. Dat werkt vergelijkbaar met een onderbrekingspunt. Het verschil is echter dat dit punt niet wordt gewist bij het afsluiten van het werkboek.



Wanneer de code gestopt is wil je misschien de waarde van de gebruikte variabelen kunnen zien. Dat kan door met de muis boven de variabelen te zweven.

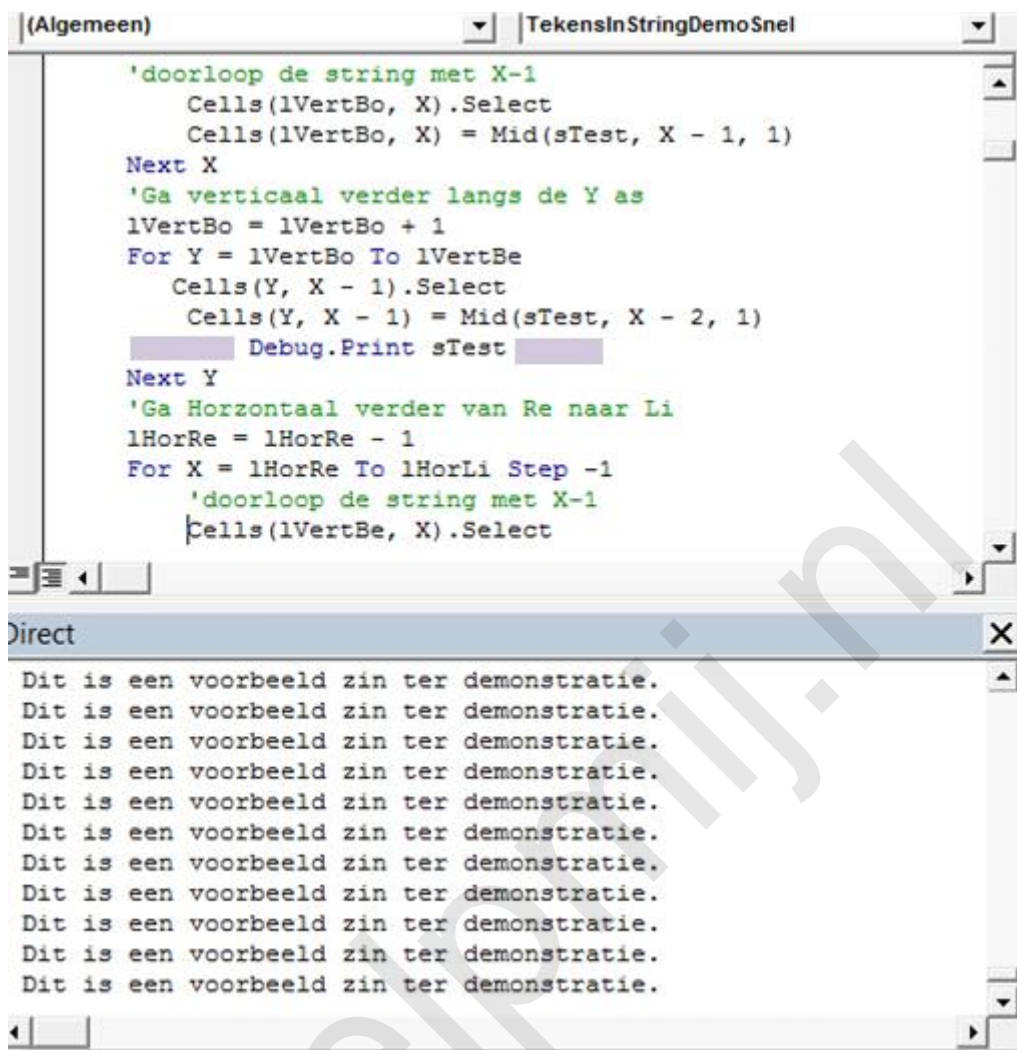


De programma uitvoering kan weer worden hervat door op F5 of F8 te drukken.

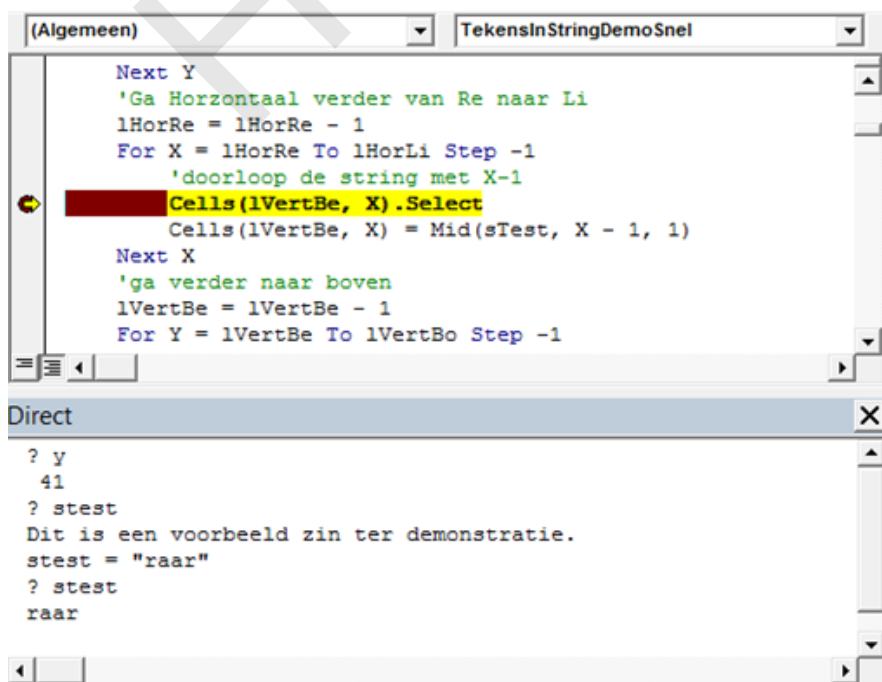
Venster Direct

Met dit venster kun je de inhoud van variabelen zichtbaar maken tijdens het uitvoeren van de code. Daarnaast kun je de waarde van de variabelen "on the fly" aanpassen. Je roept dit venster op in de Editor via Menu Beeld > Venster Direct (of met sneltoets CTRL + g). Het is weer te sluiten met het kruisje rechtsboven in dit venster.

Voor het tonen van een variabele is er het Debug.Print commando. Dat gebruik je als volgt:



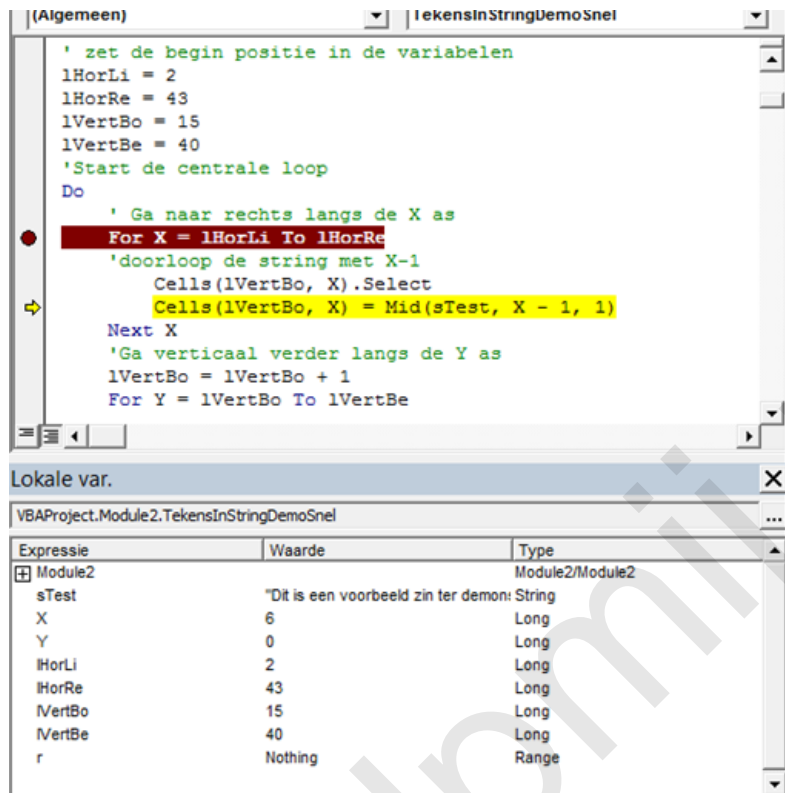
Je kunt ook de inhoud van een variabele oproepen door deze in het venster te typen, voorafgegaan door een vraagteken en een spatie:



In dit venster is het ook mogelijk om direct een andere waarde aan de variabelen toe te kennen op de manier zoals hierboven is afgebeeld (*sTest* = "raar").

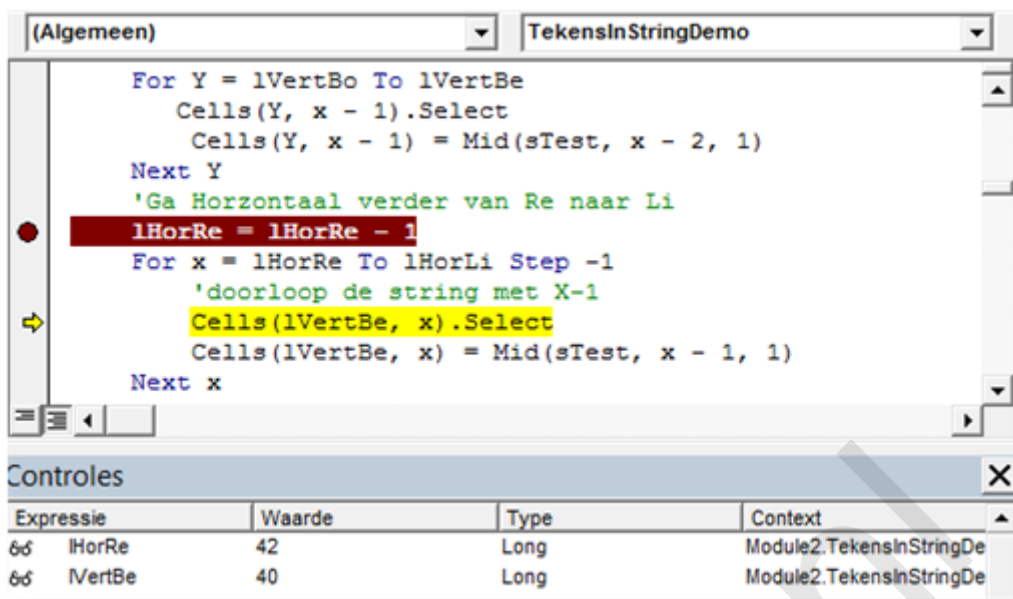
Venster Lokale Variabelen

Dit venster toont de variabelen wanneer je stap voor stap door de code gaat. Het wordt geopend via menu Beeld > Venster Lokale Variabelen. Het type en de waarde van de variabelen zijn te zien, waardoor deze eenvoudig zijn te volgen tijdens het verloop van de procedure.



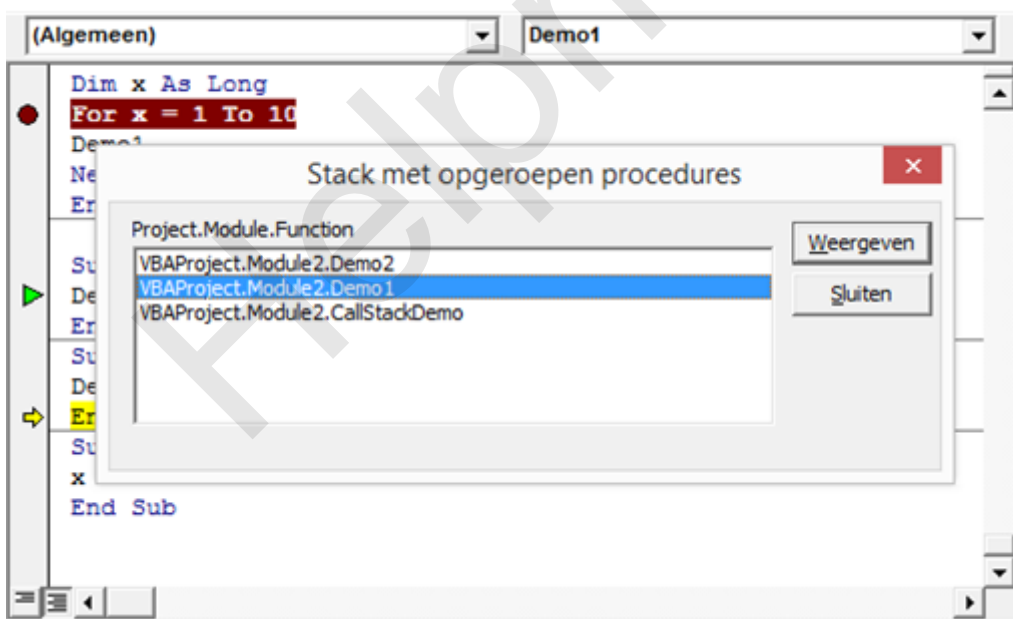
Venster Controle

Dit venster is te openen via menu Beeld > Venster Controle. Het venster maakt het mogelijk om variabelen te volgen tijdens het uitvoeren van de code. Dat kan de variabele zelf zijn, maar deze kan ook worden opgenomen in een expressie. Er zijn verschillende mogelijkheden in te stellen. Bijvoorbeeld dat de programma uitvoering wordt onderbroken als de uitkomst van een expressie waar of juist onwaar is. Ook is het mogelijk om in te stellen dat er gepauzeerd wordt bij het wijzigen van de waarde. Om dit te laten werken is het noodzakelijk dat er een controlepunt wordt ingezet. Selecteer daartoe de variabele en ga naar menu Foutopsporing > Controle. De expressie is in te vullen in het venster dat dan verschijnt. Je kunt ook op Shift + F9 drukken om een snelle controle in te stellen. Controles zijn achteraf altijd aan te passen via menu Foutopsporing > Controle bewerken of door CTRL + w te drukken. De expressies worden als volgt getoond in het Venster Controle:



De Stack met opgeroepen procedures

Dit formulier geeft weer naar welke procedures is gesprongen voorafgaand aan het huidige punt in de geselecteerde procedure. Het punt waar vandaan gesprongen werd kan worden gemarkeerd met een groen driehoekje door op Weergeven te klikken, zoals in dit voorbeeld:

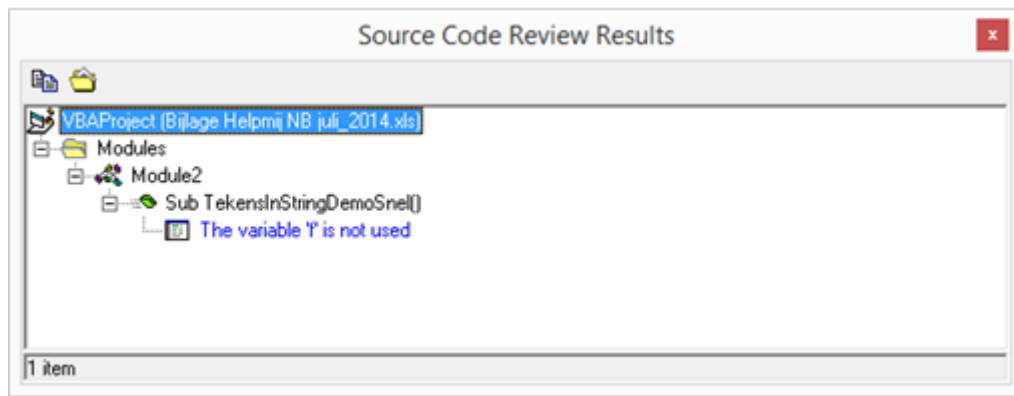


De Stack is alleen op te roepen wanneer er daadwerkelijk is gesprongen. Als dat niet het geval is wordt de menuopdracht lichtgrijs weergegeven ten teken dat de optie niet beschikbaar is.

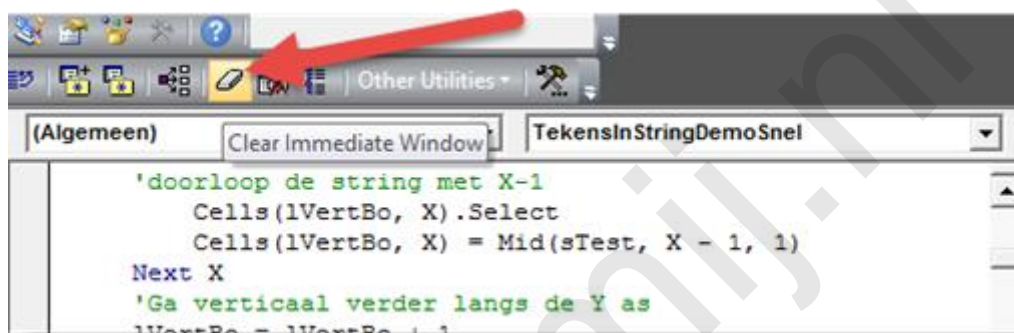
MZ-Tools

Het in een eerdere aflevering besproken MZ-Tools heeft ook twee functies die ik nog kort wil bespreken. Voor het geval je nog niet bekend bent met MZ-Tools, in [deel drie](#) is de installatie van dit tooltje besproken.

Via menu Other Utilities > Review Code wordt de code beoordeeld. Daarbij wordt onder andere aangegeven welke variabelen wel gedeclareerd zijn, maar niet werden gebruikt in de procedure. Zie dit voorbeeld:



Daarnaast is er de mogelijkheid om het venster Direct (wat behoorlijk vol kan lopen) met één klik te wissen, door op het hier getoonde gummetje te klikken:



Opschonen

Wanneer je klaar bent met de foutopsporing en je alle bugs hebt platgeslagen is het nog even zaak om alle debug commando's en onderbrekingspunten te wissen, voor je het project beschikbaar stelt aan de gebruiker. Dat is wel zo netjes.

Hiermee komt aflevering 12 tot een einde.

Samenvatting:

Fouten maken doen we allemaal. Er werden wat tips gegeven om dit zoveel mogelijk te voorkomen. Daarvoor werd onder meer een manier aangereikt om een flowchart te maken. Verder werd uitgelegd hoe je stapsgewijs door de code kan gaan en welke markeringsmogelijkheden er zijn. Daarbij werden de diverse vensters om het verloop van de code te volgen besproken. Als laatste werd de code review mogelijkheid van MZ-Tools besproken.

Volgende aflevering

In de volgende aflevering gaan we o.a. afspraken vanuit Excel (van bijvoorbeeld een actielijst) inplannen in de Outlookagenda.