VBA voor Doe het Zelvers Deel 13

Handleiding van Helpmij.nl

Auteur: leofact

December 2014

VBA voor Doe het Zelvers Deel 13

Vorige Aflevering

De vorige aflevering handelde over verschillende manieren om fouten op te sporen; het zogenaamde debuggen. Er werden mogelijkheden getoond om de inhoud van de gebruikte variabelen te tonen. Aangezien voorkomen beter is dan genezen, werden er een paar tips gegeven om een procedure meer gestructureerd en met minder kans op fouten op te zetten. Een bijlage werd bijgevoegd om te kunnen experimenteren met de besproken mogelijkheden.

In deze aflevering

Zoals beloofd in de vorige aflevering wordt dit keer het maken van een afspraak in de Outlook-agenda met behulp van Excel-VBA besproken. Hiervoor is het nodig dat er vanuit Excel contact wordt gelegd met Outlook. Dit is eerder aan bod gekomen bij het versturen van mail met behulp van Excel-VBA. Nu wordt het echter tijd om meer op de achtergrond van de verbinding tussen de verschillende Office-toepassingen in te gaan. Deze verbinding, of Binding (Engels) wordt hierna uitgelegd. Dit kan ook interessant zijn wanneer er niet van Outlook gebruik wordt gemaakt. De voorbeeldcode is uitgewerkt in een bijlage. Deze bijlage is hier te downloaden. Om deze te gebruiken is het noodzakelijk dat Outlook geïnstalleerd is. Volgende maand wordt er ingegaan op het automatiseren van Word vanuit Excel. Daarbij komt ook de Binding aan de orde en die bijlage zal dan ook bruikbaar zijn voor iedereen die over Word beschikt.

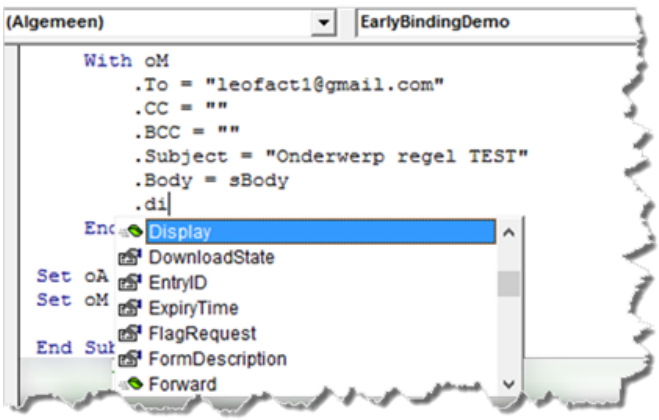
Binding?

Bij VBA wordt een verzameling objecten en eigenschappen een bibliotheek of library genoemd. Er is één algemene VBA bibliotheek Daarnaast heeft elke Office-toepassing zijn eigen specifieke bibliotheek VBA-opdrachten. Elke verschillende versie van een bepaalde toepassing heeft ook weer zijn eigen bliotheek. Wanneer je nu vanuit een bepaalde toepassing verbinding wil leggen met een andere toepassing moet er dus voor gezorgd worden dat de opdrachten herkend worden. Binding is de manier waarop Microsoft mogelijk maakt dat de ene toepassing gebruik maakt van de bibliotheek van een andere toepassing. Hierdoor kan de ene toepassing de andere aansturen. Er zijn twee manieren om deze verbinding te leggen. Namelijk Early (of vroege) Binding en Late Binding.

Early Binding

Bij Early Binding wordt de koppeling met de bibliotheek al in Design Time gelegd, dus tijdens het programmeren. Dit heeft de volgende voordelen:

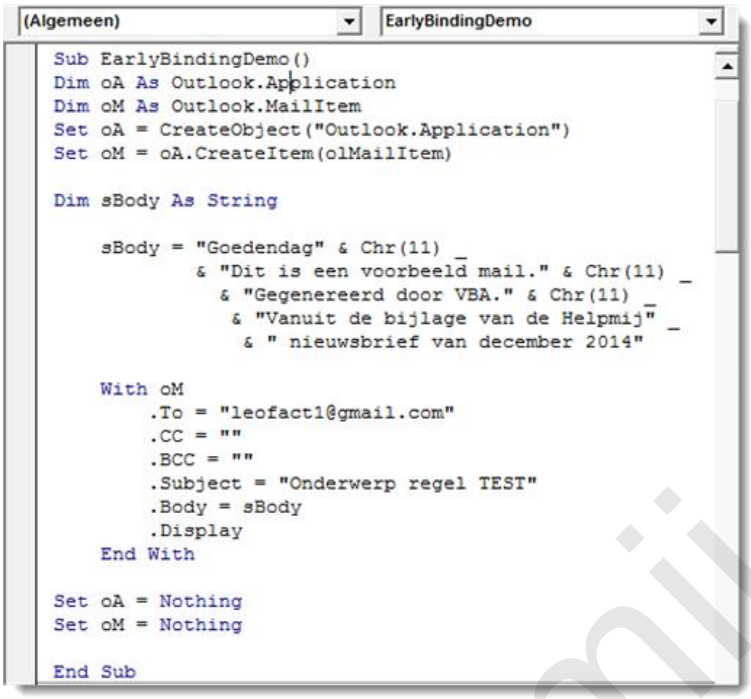
- Omdat de bibliotheek direct is gekoppeld werkt IntelliSense zoals je dat gewend bent.
- Datzelfde geldt voor alle constanten die bij de gebruikte objecten horen. Deze worden ook getoond door IntelliSense:



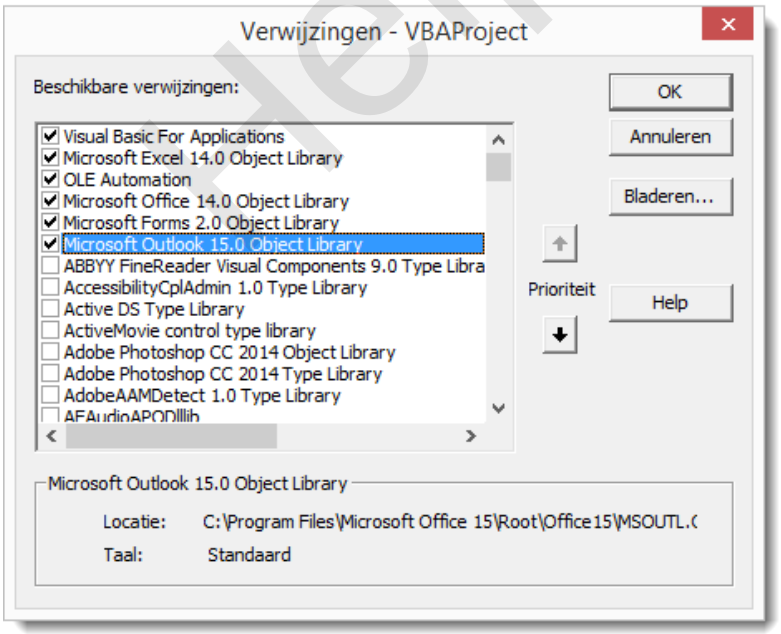
- Alle te gebruiken objecten zijn terug te vinden in de Object Browser wat het zoeken naar een bepaald object behoorlijk wat eenvoudiger maakt.
- De code wordt direct gecompileerd, waardoor dit tijdens de uitvoering niet meer hoeft te gebeuren. Dit maakt dat de macro wat sneller kan lopen.

- Doordat er direct gecompileerd wordt, worden ook de fouten direct herkend en aangegeven. Daarnaast is de foutopsporing te gebruiken waardoor het debuggen veel eenvoudiger is.

Een voorbeeld van Early Binding:



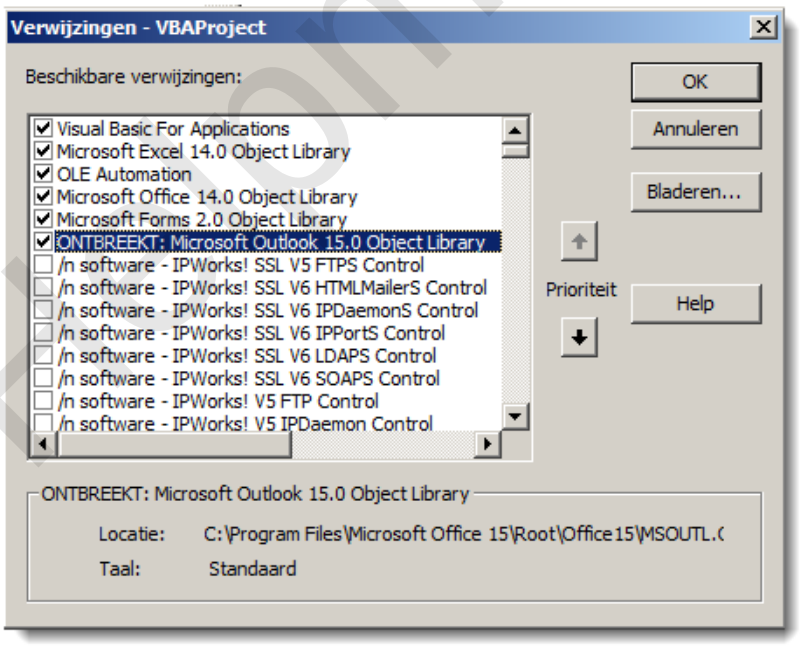
Bij Early Binding wordt de Binding gelegd tijdens het declareren van de objectvariabelen. Daarvoor moet er wel eerst een verwijzing naar de betreffende bibliotheek zijn gelegd. Dit doe je door in het venster Verwijzingen (via menu **Extra** > **Verwijzingen**) de betreffende bibliotheek aan te vinken:



De inhoud van de bibliotheek wordt bij Office voor Windows opgeslagen in een DLL (Dynamic-Link Library). Dat is de Microsoft Windows manier om programmabibliotheken op te slaan. In de aflevering over de Date and Time Picker werd uitgelegd hoe je kunt nagaan of een bepaalde DLL aanwezig is en hoe je deze eventueel kunt installeren. Deze aflevering is hier te vinden. Dit geldt helaas niet voor de bibliotheek van een nieuwere versie van een toepassing. Die is alleen te installeren door de toepassing zelf te installeren.

Early Binding heeft als groot nadeel dat het versie-afhankelijk is. Wanneer het werkboek op meerdere computers gebruikt moet worden kan het mis gaan wanneer er andere Office-versies worden gebruikt. Nieuwere versies gaat meestal nog wel, bij oudere versies gaat het echter vaak mis. Dit wordt veroorzaakt door de versie-eigen bibliotheek van het betreffende programma. Deze is meestal alleen compatible met eerdere Office-versies. Programmeer je bijvoorbeeld voor Outlook 2013, dan zal er bij gebruik van Outlook 2010 een foutmelding worden gegenereerd bij de het starten van de procedure:

In het venster Verwijzingen is dan te zien dat de bibliotheek ontbreekt:



Dit is vaak weer werkende te krijgen door de Object Library van de aanwezige Outlookversie aan te vinken. Dat wordt dan versie 14 voor Outlook 2010 en 12 voor Outlook 2007. Waarschijnlijk werkt ook de Outlook 2003 (11) bibliotheek, maar dat is hier niet getest.

Late Binding

De oplossing voor de bovengenoemde problemen is Late Binding. Hierbij wordt de binding tijdens Run Time, dus pas bij het uitvoeren gelegd. Dit is veel minder versie-afhankelijk omdat er dan verbinding wordt gelegd met de programma-eigen bibliotheek van het op de PC geïnstalleerde programma. Bij Late Binding worden de objecten niet gedeclareerd, of alleen algemeen als object. Een voorbeeld van

Late Binding:

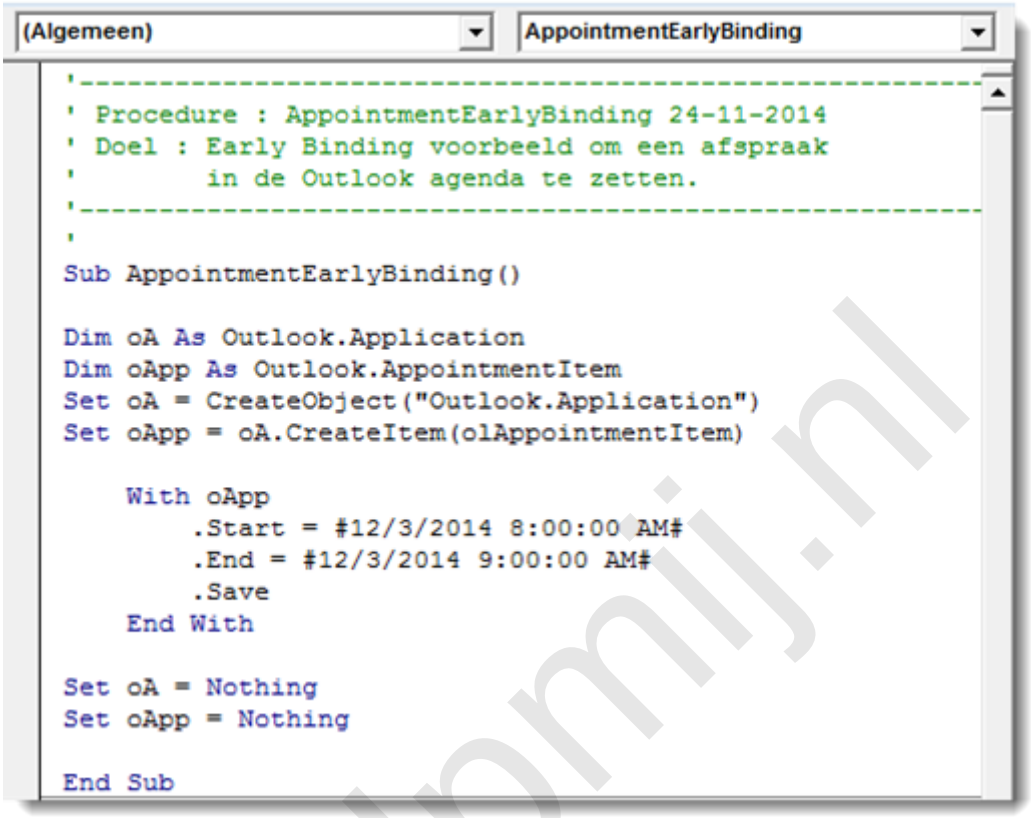
Bij Late Binding vervallen dus wel alle voordelen van Early Binding. Bij het programmeren werk je dus in “the blind”, zonder enige hulp. Pas bij het uitvoeren van de code verschijnen de eventuele foutmeldingen.

Hybride binden

Gelukkig is er met VBA voor elk probleem wel een oplossing. Het is namelijk heel goed mogelijk bij het programmeren een Early Binding te leggen om pas als de procedure helemaal af is, deze te vervangen door een Late Binding. Wanneer er bij het schrijven van de procedure alvast een beetje rekening mee wordt gehouden is dit eenvoudig uit te voeren. Je ziet duidelijk wat er moet worden aangepast wanneer je de twee eerder gegeven voorbeelden vergelijkt. Bij deze hybride oplossing wordt het programmeergemak van de Early Binding gecombineerd met de uitwisselbaarheid van de Late Binding. Denk er wel aan om ook de verwijzing weg te halen in het betreffende menu. Ontbrekende bibliotheken kunnen soms onverklaarbare fouten veroorzaken. Dat zal in dit geval niet zo'n vaart lopen, maar het is in ieder geval netter om het wel te doen.

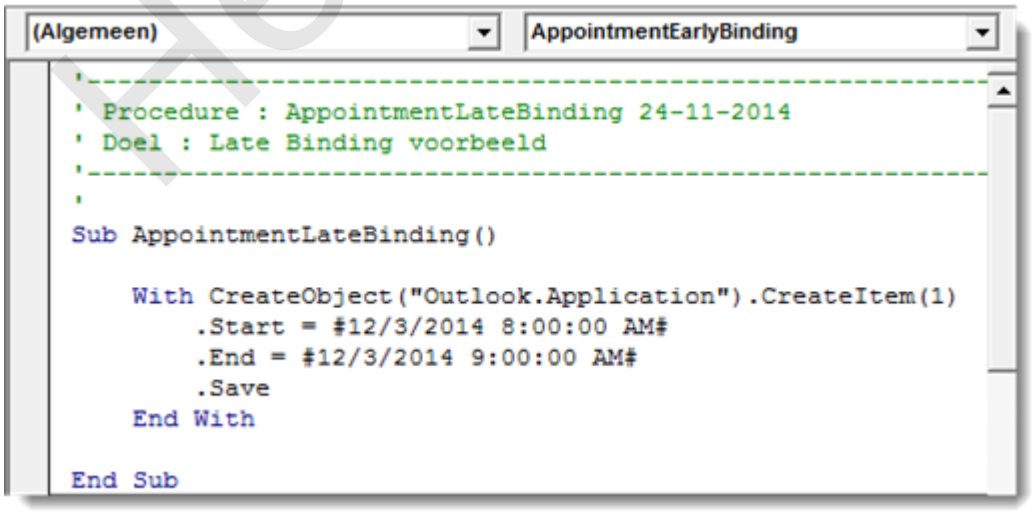
Appointment

Nu het principe van hybride programmeren bekend is, gaan we dit gebruiken om vanuit Excel een afspraak in Outlook te zetten. De basisprocedure hiervan kan er bijvoorbeeld zo uitzien:




```
'---------------------------------------------------------
' Procedure : AppointmentEarlyBinding 24-11-2014
' Doel : Early Binding voorbeeld om een afspraak
'        in de Outlook agenda te zetten.
'---------------------------------------------------------
'
Sub AppointmentEarlyBinding()

Dim oA As Outlook.Application
Dim oApp As Outlook.AppointmentItem
Set oA = CreateObject("Outlook.Application")
Set oApp = oA.CreateItem(olAppointmentItem)

    With oApp
        .Start = #12/3/2014 8:00:00 AM#
        .End = #12/3/2014 9:00:00 AM#
        .Save
    End With

Set oA = Nothing
Set oApp = Nothing

End Sub
```

De procedure kan als volgt omgebouwd worden om een Late Binding te maken:




```
'---------------------------------------------------------
' Procedure : AppointmentLateBinding 24-11-2014
' Doel : Late Binding voorbeeld
'---------------------------------------------------------
'
Sub AppointmentLateBinding()

    With CreateObject("Outlook.Application").CreateItem(1)
        .Start = #12/3/2014 8:00:00 AM#
        .End = #12/3/2014 9:00:00 AM#
        .Save
    End With

End Sub
```

Nu wordt Outlook op de achtergrond geopend met CreateObject en de afspraak wordt aangemaakt met CreateItem. Omdat er tijdens Design Time geen verbinding is met de Outlook Bibliotheek kan er geen gebruik gemaakt worden van de namen van de Outlook constanten. Er moet gebruik worden gemaakt van de waardes hiervan. Vervang daarom olAppointmentItem door de waarde waar dit voor staat; namelijk 1. Doe je dit niet dan volgt tijdens de uitvoering een fout 438; ”Deze eigenschap of methode wordt niet ondersteund door dit object.”

Deze procedure blijft prima werken wanneer je de eerder gelegde verwijzing naar de Outlook bibliotheek weer weghaalt (door het vinkje weg te halen in het venster Verwijzingen). De procedure zal nu op vrijwel alle pc's draaien waarop Outlook aanwezig is.

Er staat nu dus een afspraak in de Outlook-agenda. De gebruikte procedure heeft echter nog behoorlijk wat beperkingen. De afspraak komt alleen in de standaardagenda en de tijd moet in het Engelse formaat worden opgegeven. Kortom, er is nog veel meer mogelijk. Daarom schakelen we weer terug naar Early Binding. We leggen de Outlook-verwijzing weer vast in het verwijzingenvenster en starten met de Early Binding basisprocedure. Deze breiden we vervolgens als volgt uit:

(Algemeen) | OutlookAppLateBinding2

```
Option Explicit
'---------------------------------------------------------------------
' Procedure : OutlookAppointment 24-11-2014
' Doel : Early Binding: Afspraak maken mbv Excel VBA
'---------------------------------------------------------------------
'
Sub OutlookAppEarlyBinding()

Dim oAp As Outlook.Application
Dim oApp As Outlook.AppointmentItem
Dim oNs As Outlook.Namespace
Dim oAgenda As Outlook.MAPIFolder
Dim oSubAgenda As Outlook.MAPIFolder
Dim sAgenda As String
Set oAp = Outlook.Application

Set oNs = oAp.GetNamespace("MAPI")
'Stel de standaard agenda in
Set oAgenda = oNs.GetDefaultFolder(olFolderCalendar)
'## gebruik de volgende code om een sub agenda in te stellen
'   (dit moet een bestaande zijn)
'Set oSubAgenda = oAgenda.Folders("Agenda")
Set oApp = oAgenda.Items.Add(olAppointmentItem)
'Gebruik voor een sub agenda:
'Set oApp = oSubAgenda.Items.Add(olAppointmentItem)

    With oApp
        'geef de afspraak eigenschappen
        .Subject = "vergadering"
        .Location = "Vergaderzaal 0.16"
        .Body = "vergadertekst"
        .BusyStatus = olBusy
        .Start = DateValue("10-12-2014") + TimeValue("9:00:00")
        .End = DateValue("10-12-2014") + TimeValue("10:00:00")
        .Subject = "vergadering"
        .Location = "Vergaderzaal 0.16"
        .Body = "vergadertekst"
        .BusyStatus = olBusy
        .AllDayEvent = False
        .ReminderMinutesBeforeStart = 15
        .ReminderSet = True
        .Categories = "rood"
        .Display 'save
      End With

Set oApp = Nothing
Set oAp = Nothing

End Sub
```


Nu worden er veel meer eigenschappen van de afspraak ingesteld. Dankzij IntelliSense en de Object Browser is het nu vrij eenvoudig om zelf de gewenste eigenschappen te zoeken en te gebruiken. Verdere uitbereiding wordt dan ook aan de eigen fantasie overgelaten. Voor verspreiding is het nu zaak om de procedure naar Late Binding over te zetten. Dat kan door de objecten algemeen te maken

en door de namen van de constanten te vervangen door de waardes die ze vertegenwoordigen. Dat kan er dan zo uitzien:

(Algemeen) | OutlookAppLateBinding

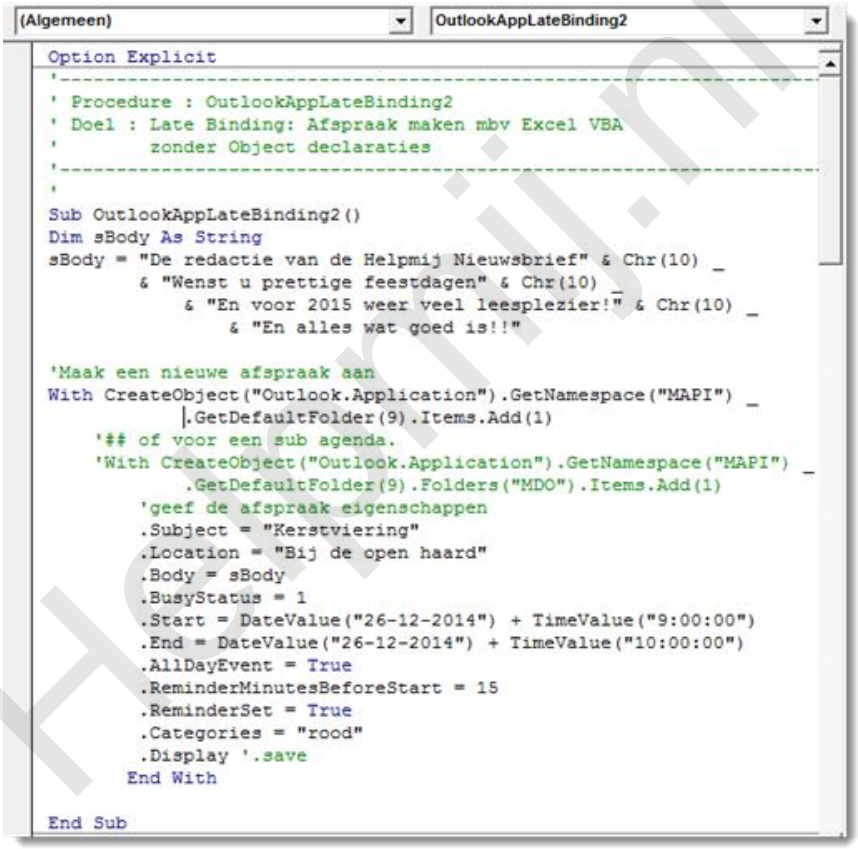
```
'---------------------------------------------------------------------
' Procedure : OutlookAppointment 24-11-2014
' Doel : Late Binding: Afspraak maken mbv Excel VBA
'---------------------------------------------------------------------
'
Sub OutlookAppLateBinding()

Dim oAp As Object
Dim oApp As Object
Dim oNs As Object
Dim oAgenda As Object
Dim oSubAgenda As Object
Dim sAgenda As String
Dim sBody As String
sBody = "De redactie van de Helpmij Nieuwsbrief" & Chr(10) _
        & "Wenst u prettige feestdagen" & Chr(10) _
            & "En voor 2015 weer veel leesplezier!" & Chr(10) _
                & "En alles wat goed is!!"

'Open Outlook op de achtergrond.
Set oAp = CreateObject("Outlook.Application")
Set oNs = oAp.GetNamespace("MAPI")
'Stel de standaard agenda in
Set oAgenda = oNs.GetDefaultFolder(9)
'## gebruik de volgende code om een sub agenda in te stellen
'   (dit moet een bestaande zijn)
'Set oSubAgenda = oAgenda.Folders("Agenda")
Set oApp = oAgenda.Items.Add(1)
'Gebruik voor een sub agenda:
'Set oApp = oSubAgenda.Items.Add(1)


    With oApp
        'geef de afspraak eigenschappen
        .Subject = "Kerstviering"
        .Location = "Bij de open haard"
        .Body = sBody
        .BusyStatus = 1
        .Start = DateValue("26-12-2014") + TimeValue("9:00:00")
        .End = DateValue("26-12-2014") + TimeValue("10:00:00")
        .AllDayEvent = True
        .ReminderMinutesBeforeStart = 15
        .ReminderSet = True
        .Categories = "rood"
        .Display '.save
       End With

Set oApp = Nothing
Set oAp = Nothing
Set oNs = Nothing
Set oAgenda = Nothing

End Sub
```

Het vervangen van de constanten kan nog een aardige uitdaging vormen. Hoe weet je immers al die waardes. De Microsoft-help is hierin helaas niet zo heel behulpzaam. Gelukkig zijn er wel sites te vinden waar die waardes zijn te vinden. Bijvoorbeeld hier te downloaden als zip bestand. Dat is natuurlijk mooi, maar ook een heel gezoek. Gelukkig zijn er nog twee ontsnappingsroutes mogelijk om snel achter de waarde te komen. Dit moet wel worden uitgevoerd in de routine wanneer de Early Binding nog bestaat. Als je de routine pauzeert tijdens het uitvoeren (zie aflevering 12) zie je de

waarde wanneer je met de muis over de naam van de constante "hoovert" (er boven laten zweven). Een andere manier is om de waarde in het venster Direct op te zoeken:

Ook over het gebruik van dit venster is meer te vinden in aflevering 12.

Er is nog een andere manier om een Late Binding te leggen. Deze vergt iets meer werk om de Early Binding in een Late Binding om te zetten. Het voordeel is echter dat er geen objecten gedeclareerd hoeven te worden. Dit kan als volgt:



Met dit voorbeeld eindigt aflevering 13.

Samenvatting:

Besproken is de manier waarop een Office-toepassing via VBA verbinding kan maken met een andere Office-toepassing. Daarbij is uitgelegd dat dit gaat om de verbinding met de object bibliotheken (of Libraries) van de betreffende toepassingen. De manier van verbinden is aan de hand van de begrippen Early Binding en Late Binding uitgelegd. Hiervan zijn de voor- en nadelen opgesomd en is aangegeven hoe hier het beste mee kan worden omgegaan. Als laatste is er een uitgewerkte afspraak gemaakt en werd deze op twee manieren omgezet naar een Late Binding. Dit met een paar tips om dat eenvoudiger te maken.

Volgende aflevering

De volgende keer gaan we verder met de Binding en gaan dan met name met Word aan de slag. Er wordt o.a. een tabel in Word aangemaakt met behulp van Excel-VBA.