VBA voor Doe het Zelvers deel 19

Handleiding van Helpmij.nl

Auteur: leofact

Juni 2015

“ Dé grootste en gratis computerhelpdesk van Nederland ”

Vorige aflevering

In deel 18 was te lezen hoe werkboekbeveiliging kan worden ingesteld. Dit om de inhoud te beschermen tegen ongewenste pottenkijkers en om de userinterface en opmaak intact te houden tijdens allerlei vormen van gebruik van het werkboek. Naast de normale methode van instellen via het menu werd ook steeds de betreffende VBA-code gegeven.

Deze aflevering

In deel 19 wordt dit vervolgd met beveiligingsmethodes om de userinterface en opmaak intact te houden tijdens allerlei vormen van gebruik van het werkboek. Daarnaast worden meerdere vormen van beveiliging tegen ongewenst bewerken gegeven. Ook in deze aflevering wordt naast de normale methode van instellen via het menu tevens de betreffende VBA-code gegeven. Deze is weer te vinden in de bijlages die hetzelfde zijn als bij de vorige aflevering; namelijk een office 2003 .xls werkboek en een Office 2007+ .xlsm werkboek. De werkboeken verschillen alleen qua office-versie.

Werkmapbeveiliging

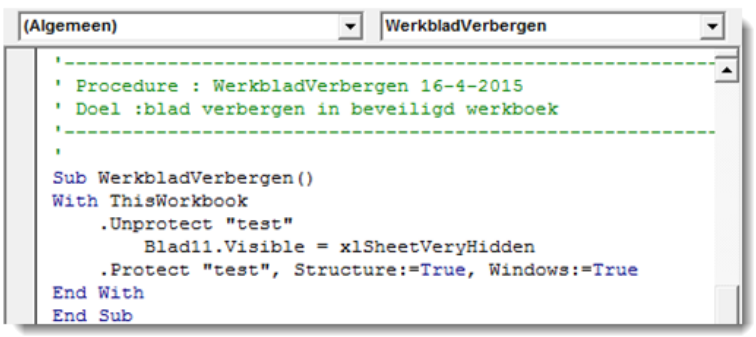
Met deze beveiliging kan worden voorkomen dat gebruikers werkbladen verwijderen, toevoegen of verbergen (structuur). Daarnaast kunnen de geopende vensters tegen aanpassen beschermd worden. Op deze manier kan een bepaald werkboek steeds op uniforme wijze aan de gebruiker worden aangeboden, wat het gebruiksgemak kan verhogen. De beveiliging is in te stellen via de tab **Controleren** > **Werkmap beveiligen.** Vink in het betreffende venster aan of de structuur en/of de vensters beveiligd moeten worden. Dit kan worden vastgelegd met een wachtwoord. Dat is echter niet verplicht. In het laatste geval kan de gebruiker de beveiliging ongedaan maken. In een enkel geval kan dat juist wenselijk zijn, afhankelijk van het doel wat de ontwikkelaar voor ogen heeft. Werkmap beveiliging is ook met VBA in te stellen:

```
ThisWorkbook.Protect "test", Structure:=True, Windows:=True
```

De beveiliging is als volgt op te heffen.

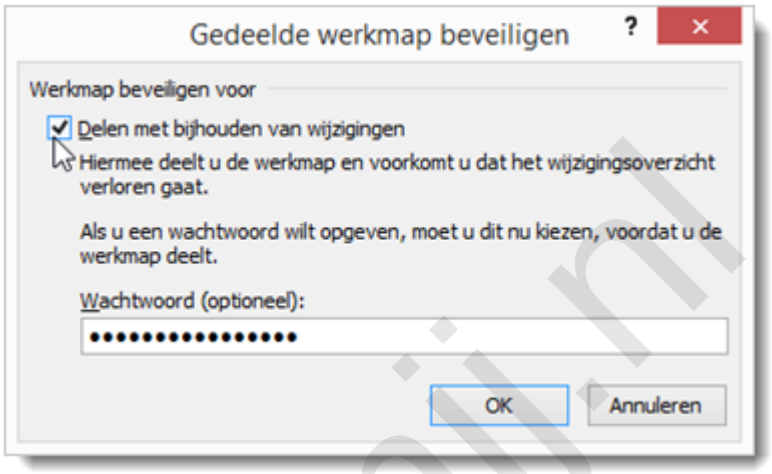
```
ThisWorkbook.Unprotect "test"
```

Dit laatste kan nodig zijn wanneer er werkbladen gemanipuleerd moeten worden. Bijvoorbeeld: om een werkblad te verbergen hef je eerst de beveiliging op, verberg je vervolgens het werkblad, om daarna de beveiliging er weer op te zetten. Dat kan dan zoiets worden:



Gedeelde werkmap beveiligen

In deel 16 is het delen van een werkboek behandeld. Het delen en bijhouden van wijzigingen kan daarbij eenvoudig worden uitgezet door de gebruikers. Soms is dit niet gewenst. In dat geval kan het delen en bijhouden van wijzigen beveiligd worden met een wachtwoord. Dat doe je via de tab **Controleren** > **Werkmap beveiligen en delen.** Zie daarvoor het volgende screenshot:



Bij gebruik van VBA gaat dat als volgt:

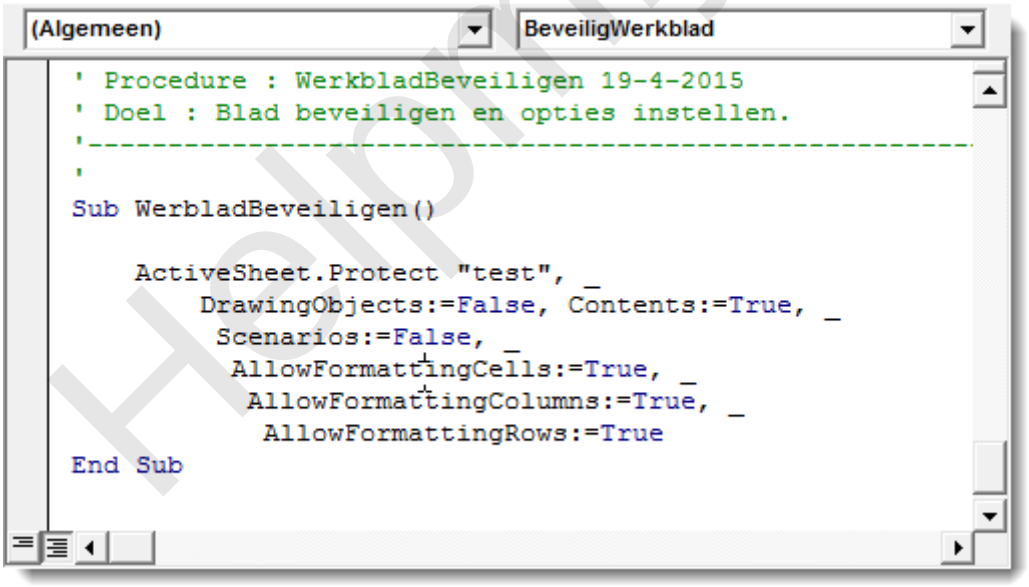
Let op: Het is normaal dat de code niet is te zien in een gedeelde werkmap. De code kan pas weer worden bewerkt wanneer het delen is uitgezet. Dat gaat met de dezelfde knop; deze heet nu **Beveiliging gedeelde werkmap opheffen.**

Werkblad beveiligen

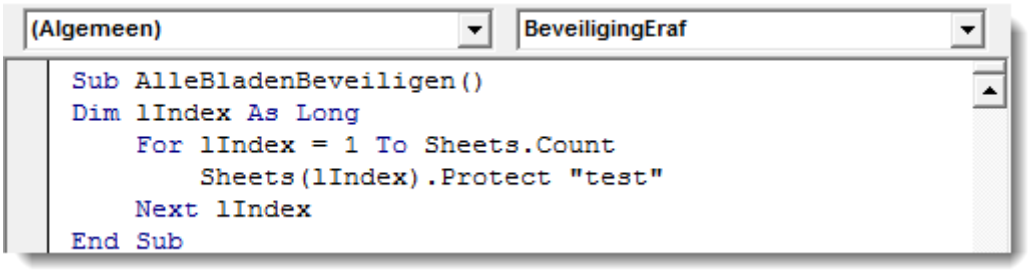
Zoals bekend zijn de werkbladen te beveiligen tegen bepaalde bewerkingen. Dit kan via de tab **Controlere**n > **Blad beveiligen**.

In het betreffende venster vul je dan het wachtwoord in (optioneel) en vink je vervolgens aan wat je wilt toestaan:

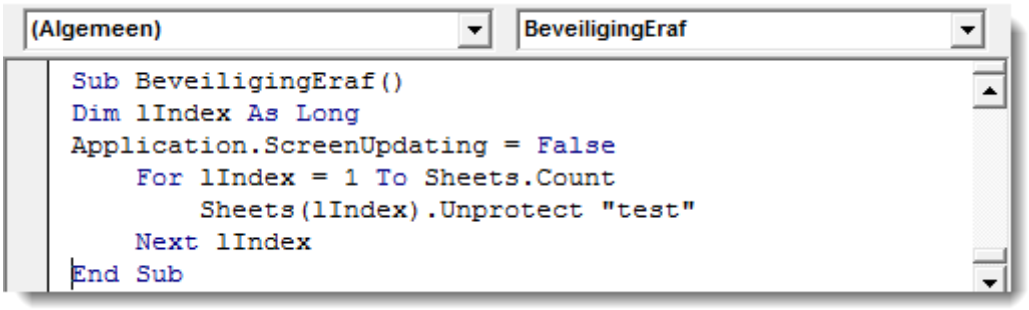
Opheffen van de beveiliging doe je met dezelfde knop, die nu **Beveiliging blad opheffen** heet. Met behulp van VBA gaat het beveiligen van een blad als volgt:



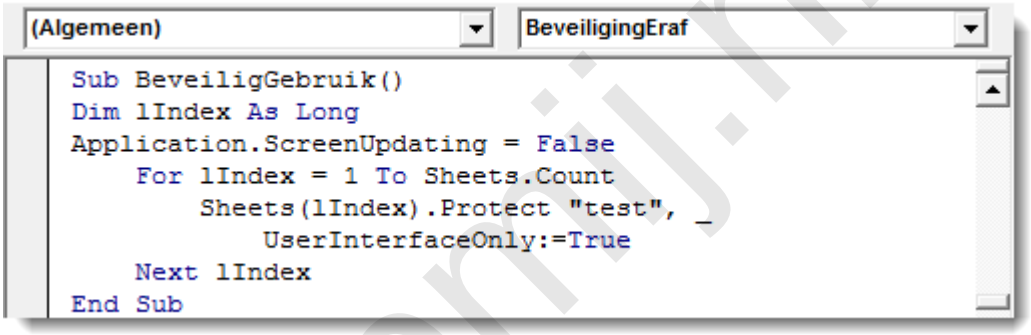
Met VBA is het eenvoudig om alle bladen in één keer te beveiligen:



Met een hele kleine wijziging wordt de beveiliging weer opgeheven:



Met behulp van VBA is er nog één bijzondere extra optie mogelijk bij het beveiligen van een werkblad. Het is namelijk mogelijk om een werkblad te beschermen tegen veranderingen door de gebruiker, terwijl aanpassingen die door VBA worden gemaakt wél worden toegestaan. Dat gaat als volgt:



Let op: bij gedeelde werkboeken kan de beveiliging niet worden aangepast. Er volgt een foutmelding als dat toch wordt geprobeerd.

Met de eigenschap Protection kan worden nagegaan of bepaalde beveiliging actief is. Met de volgende code is te achterhalen of in de beveiliging het toevoegen van kolommen is toegestaan:

```
MsgBox ActiveSheet.Protection.AllowInsertingColumns
```

Nota bene: dit geeft alleen weer of het invoegen van kolommen is aangevinkt. Het invoegen van kolommen wordt pas daadwerkelijk geblokkeerd wanneer de beveiliging van het werkblad actief is.

De beveiliging van werkbladen is vrij simpel met behulp van VBA op te heffen, ook wanneer het wachtwoord niet bekend is. Op internet zijn er methodes te vinden waarbij in no time achterhaald wordt welk wachtwoord werkt om de beveiliging op te heffen. Helpmij staat terecht niet toe dat er methodes worden gegeven om wachtwoorden te kraken. De procedure geef ik dan ook niet. Overigens is deze beveiliging verbeterd in Excel 2013 en voor zover mij bekend, werken de genoemde procedures daarin niet meer.

Celblokkering

De cel is het kleinste object wat beveiligd kan worden. De beveiliging is ook hier afhankelijk van de werkbladbeveiliging. Of een cel in een beveiligd werkblad kan worden bewerkt geef je aan door deze te blokkeren met behulp van de celeigenschappen (Druk *CTRL + 1*; **Celeigenschappen** > tab **Bescherming** > **Blokkeren).** In VBA:

```
Blad1.Cells.Locked = True
```

De blokkering wordt opgeven door de waarde False te geven. Dan kan de inhoud van de cel weer worden bewerkt.

VBA beveiligen

Het laatste onderdeel wat wordt besproken is de beveiliging van de code zelf in de VB-editor. Dat stel je in via menu **Extra** > **Eigenschappen van VBAProject…** > **Vergrendelen tegen weergave**. Door hier een wachtwoord in te voeren is het voor de gebruiker niet mogelijk om de code in te zien of aan te passen. Vooral dat laatste kan een hoop problemen voorkomen. Ook deze beveiliging is relatief eenvoudig te kraken. Bij gemiddeld gebruik is het echter voldoende om het project te beschermen tegen ongewenste aanpassingen. De project-vergrendeling is in principe niet met VBA in te stellen, al ben ik wel een procedure tegen gekomen waarbij Sendkeys werd gebruikt. Deze methode, waarbij de toetsaanslagen worden gesimuleerd, werkt echter niet echt failsafe.

Samenvatting

In dit deel werd besproken hoe je de structuur en vensteropbouw van een werkmap kunt beveiligen. Ook werd behandeld hoe je een gedeelde werkmap kunt beveiligen tegen het wissen van de bijgehouden wijzigingen. Daarnaast werden er diverse manieren aangeboden om werkbladen te beveiligen. Hieronder een methode om het werkblad te beveiligen tegen wijzigingen door de gebruiker, terwijl wijzigingen door VBA wél mogelijk blijven. Er werd daarbij kort aangestipt hoe kan worden opgevraagd welke handelingen er worden toegestaan door de werkbladbeveiliging. Daarnaast kwam ook de celblokkering aan de orde. Als laatste werd besproken hoe het project zelf kan worden beveiligd.

Deel 20

In de volgende aflevering wordt het gebruik van operators en expressies besproken. Dit om te kunnen rekenen en vergelijken.

Helpmij.nl