



Access voor beginners Hoofdstuk 19 (2)

Handleiding van Helpmij.nl

Auteur: OctaFish

Mei 2014

“ Dé grootste en gratis computerhelpdesk van Nederland ”

Verlaat de database!

Een beheerder die met een grote groep gebruikers in dezelfde database werkt, kent dat vast wel: je moet onderhoud plegen op de database, maar er is altijd wel een gebruiker die de database open heeft staan, zodat je niet bij de tabellen kunt.

Uiteraard zet je dan een berichtje op het intranet, maar eens te meer blijkt dat hooguit 5% van de medewerkers die pagina's regelmatig leest... Dus stuur je alle gebruikers maar weer een mailtje, maar dan blijkt dat er één een dag vrij is, dus je krijgt een AutoReply waar je niks aan hebt, en een tweede is net naar een vergadering op het stadhuis, en heeft de database open laten staan, en een derde heeft de pc de vorige dag aan laten staan, en zit nu ziek thuis. Kortom: de DB is door een aantal gebruikers gelockt, en je krijgt ze niet te pakken.

De vraag is dus: hoe krijg je, op het moment dat jij dat wilt, de gebruikers uit je database?

Daar zijn een aantal manieren voor te vinden. Zo kun je in de bedrijfspolicy opnemen dat alle computers 's nachts automatisch worden afgesloten, zodat de DB in ieder geval de volgende dag weer schoon is. Maar daar heb je overdag weinig aan. Bij iedereen persoonlijk langs gaan wil meestal wel helpen, maar kost tijd, en wat doe je als de gebruiker (zie boven) niet te bereiken is?

Je wilt kortom een nette manier om de gebruiker op afstand uit de database te krijgen. In dit hoofdstuk leg ik daarvoor twee methoden uit: registreren van gebruikersactiviteit, en op basis daarvan een actie uitvoeren (database sluiten), en een methode om alle geopende databases automatisch te sluiten.

Beide methoden zijn wel vrij drastisch, zoals bij geautomatiseerde processen gebruikelijk is. Automatische slagbomen gaan ook niet in gesprek met de aanstormende auto's, die gaan dicht als de trein er aan komt. En de onderstaande procedures doen dat dus ook.

Database automatisch sluiten

De meest 'starre' variant is er een waarbij de beheerder van de DB in een systeemtabel een specifieke waarde vastlegt, waardoor alle systemen automatisch afsluiten. Deze waarde wordt door een startformulier continue uitgelezen uit de database, en zodra de waarde verandert, wordt de afsluitprocedure gestart.

Laten we de tabel voor het gemak [Settings] noemen. Hierin maak je 3 velden:

- Logoff (Ja/Nee veld)
- LogOffTime (<Datum/Tijd> met <Lange Tijdnotatie>)
- RestartTime (<Datum/Tijd> met <Lange Tijdnotatie>)

Zo'n tabel ziet er bijvoorbeeld zo uit:

	Logoff	LogOffTime	RestartTime
▶	<input type="checkbox"/>	0:02:00	0:10:00
*	<input type="checkbox"/>		

Het veld [Logoff] gaan we gebruiken om de actie te triggeren. De waarde *Nee* betekent dan dat de DB gewoon gebruikt kan worden. Veranderen we die straks in *Ja*, dan wordt de database automatisch afgesloten. De tijd waarop dat gebeurt bepalen we middels het veld [LogOffTime]. In het voorbeeld is dat 2 minuten. Ga je het systeem echt gebruiken, dan is dat natuurlijk best kort, en geeft een tijdsduur van een kwartier of meer de gebruikers voldoende tijd om hun werk (en de DB) af te sluiten.

Het veld [RestartTime] is in beginsel niet nodig voor de werking van het systeem, en wordt slechts gebruikt om de gebruikers een idee te geven van wanneer de DB weer beschikbaar is.

Als de procedure wordt gestart door het veranderen van het veld [Logoff], krijgt de gebruiker het volgende scherm te zien:



De tijd waarop de DB dicht gaat komt hier uit het veld [LogOffTime], en de tijd waarop de DB weer beschikbaar is wordt berekend op basis van het veld [RestartTime] in combinatie met [LogOffTime] en de functie Now(). De gebruikte code hiervoor wordt verderop behandeld.

Bovenstaand formulier is het *resultaat* van de controle, maar is niet het formulier dat de controle *uitvoert*. Daarvoor hebben we een ander formulier nodig. Dat bespreken we in paragraaf *Het formulier frmExitNow*.

Het formulier frmLogOff

Het formulier dat de code controleert noemen we voor het gemak <frmLogOff>, en het bevat als belangrijkste onderdeel de code die nodig is om de afsluitprocedure uit te voeren. Dit formulier maakt gebruik van de *Timer* functie van een formulier. Elk formulier kent de gebeurtenissen <Bij Timer> (OnTimer) en <Timerinterval>. Standaard staat bij *Timerinterval* de waarde 0, wat betekent dat er geen timer actief is. Zet je daar een waarde neer die groter is dan 0, dan reageert het formulier op basis van de ingestelde tijdwaarde. Het formulier kan dan bijvoorbeeld elke seconde, of elke minuut een actie uitvoeren. Die actie leg je vast in de gebeurtenis <Bij Timer>. Stel je dus alleen een tijd in, maar maak je geen gebeurtenis, dan gebeurt er nog steeds niet zo heel veel.

Het formulier <frmLogOff> hoeft verder helemaal niks te doen, en er hoeft ook helemaal niks op te staan. Dus je hoeft er geen tabel aan te koppelen of er tekstvelden op te zetten. Al mag dat natuurlijk wel, want dan kun je des te makkelijker de waarden uit de tabel lezen. Als het formulier *niet* is gekoppeld, moet je de waarden met een Recordset ophalen. Het formulier is, voor het gemak, wèl aan de tabel [Settings] gekoppeld. In dit hoofdstuk gebruik ik beide varianten; het hierboven getoonde formulier in de afbeelding is daarom niet aan een tabel gekoppeld. Op die manier kun je zelf bepalen welke variant voor jou het beste werkt.

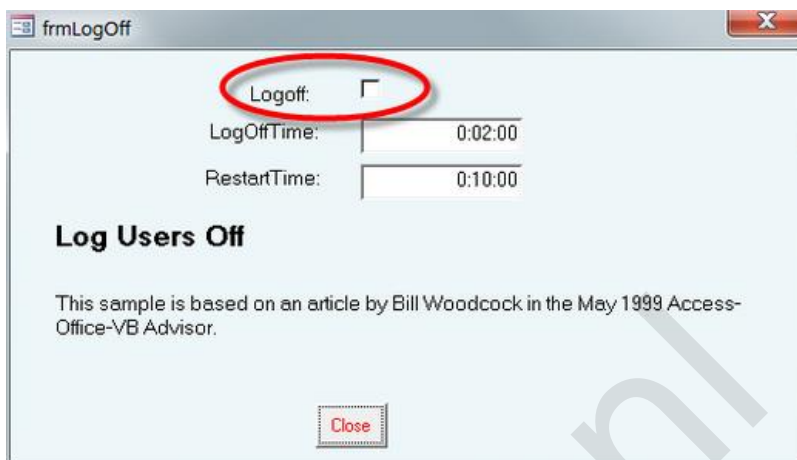
Wat moet het formulier doen?

Om correct te werken, moeten de volgende zaken worden geregeld:

- Gebeurtenis <Bij Laden> van het formulier

- Gebeurtenis <Bij Timer> van het formulier
- Formulier koppelen aan tabel [Settings]

Laten we eens kijken wat er gebeurt op het formulier.



Het formulier bevat een koppeling met de tabel [Settings], en je kunt de velden uit die tabel dus op het formulier laten zien. En dus ook op het formulier aanpassen. Dat laatste lijkt handig, maar je moet dan natuurlijk wel voorkomen dat de gewone gebruiker er niet bij kan, want dan kan de gebruiker niet alleen de tijden aanpassen, maar zelfs het proces onderbreken door het vinkje weer uit te zetten. En als het vinkje uit staat, wordt de afsluitprocedure niet uitgevoerd.

Een manier om te voorkomen dat onbevoegden het formulier zien of kunnen bewerken, is om er toegangsniveaus aan te hangen, die je in een User tabel bijhoudt. Een gebruiker met het verkeerde toegangsniveau kan dan niets uitrusten op het formulier, en de beheerders, die wel de juiste rechten hebben uiteraard wel.

Bij het laden van het formulier wordt de volgende code uitgevoerd:

```
Private Sub Form_Load()
If Me.Logoff Then
    MsgBox "De database is gesloten voor onderhoud." & vbCrLf & vbCrLf
        & "Probeer het later nog eens."
    Application.Quit
End If
DoCmd.Hourglass False
End Sub
```

Er wordt gekeken naar de waarde van het veld LogOff om te bepalen wat er moet gebeuren. Als de waarde is ingesteld (aangevinkt dus) dan wordt de database direct gesloten. Er komt nog wel een msgbox, maar daarna is het over en uit voor de gebruiker.

Interessanter is de functie Timer.

Selectietype: Formulier	
Formulier	
Opmaak	Gegevens
Gebeurtenis	Overige
Alle	
Bij timer	[Gebeurtenisprocedure]
Timerinterval	5000

De gebeurtenis zie je hieronder, de waarde *Timerinterval* is een getal die je moet intypen. Op het formulier zie je de waarde 5000, en dat zal duidelijk maken dat we hier niet meten in secondes. De waarde 5000 is dan ook exact 5 seconden, wat gelijk duidelijk maakt dat je de gewenste tijd in seconden met 1000 moet vermenigvuldigen. De rekenfactor is dus 1000, en dat staat voor 1 seconde. Wil je de procedure elke minuut laten uitvoeren, dan moet je de waarde 60.000 intypen, zijnde $60 * 1000$. En voor 5 minuten gebruik je dus de waarde $5 * 60 * 1000 = 300.000$.

De procedure <Bij timer> bevat de volgende code:

```
Private Sub Form_Timer()

Dim Msg As String, intLogoff As Integer

intLogoff = Me.Logoff

If intLogoff = True Then

    If Me.Tag = "MsgSent" Then

        Application.Quit (acQuitSaveAll)

    Else

        Me.Tag = "MsgSent"

        DoCmd.OpenForm "frm_ExitNow"

    End If

End If

End Sub
```

Ik heb de eigenschap <Extra Info> (<Tag> in het Engels) vaker gebruikt, en ook hier komt hij weer prima van pas. De procedure wordt dus uitgevoerd met het interval dat is ingesteld bij <Timerinterval>. En dan wordt er dus weer gekeken naar het veld Logoff, dat de waarde TRUE of FALSE geeft.

De gebruiker wordt niet gelijk als het vinkje wordt gezet uit de db gegooid; eerst krijgt, na het aanvinken, de eigenschap TAG een vaste tekst ('MsgSent'). De volgende keer dat de routine wordt uitgevoerd, en de eigenschap TAG dus de vaste tekst bevat, wordt de database gesloten met de opdracht Application.Quit (acQuitSaveAll). De parameter zorgt ervoor dat eventuele openstaande tabellen of queries waarin records zijn bewerkt worden opgeslagen, zodat er geen gegevens verloren raken.

Het formulier frmExitNow

Het formulier frmExitNow is het formulier dat we tonen op het scherm als de afsluitprocedure wordt gestart. Tenslotte moeten we de gebruiker nog wél mededelen dat de database dicht gaat voor onderhoud. Dit formulier heeft als belangrijke eigenschap dat de optie *Modaal* op **Aan** staat. We zorgen er daarmee voor dat zolang het formulier open staat, er in de database niets anders gedaan kan worden. Je zult dus eerst iets moeten doen met het formulier.

Ook hier zit wat code in verwerkt. De volgende procedures worden op het formulier uitgevoerd: Form_Load en Form_Timer. De procedure Form_Load is vrij lang, dus die knip ik in delen.

```
Private Sub Form_Load()
```

```
Dim snp As Recordset
Dim strText As String, sTimeBlock As String, sNewTimeBlock As String
Dim iTime As Integer, iNewTime As Integer
Static myNewStartTime As Variant, myTime As Variant
```

Er zijn nogal wat variabelen nodig, omdat de tijd dynamisch op het formulier wordt berekend. De meeste variabelen zijn normaal, maar de variabele myNewStartTime wordt als Static gedefinieerd. Die hoeft namelijk niet te veranderen.

```
Set snp = CurrentDb.OpenRecordset("Settings", dbOpenSnapshot)
iTime = Minute(snp!LogOffTime)
iNewTime = Minute(snp!RestartTime)
If iTime = 0 Then
    iTime = Second(snp!LogOffTime)
    sTimeBlock = "seconden"
ElseIf iTime = 1 Then
    sTimeBlock = "minuut"
Else
    sTimeBlock = "minuten"
End If
If iNewTime = 0 Then
    iNewTime = Second(snp!RestartTime)
    sNewTimeBlock = "seconds"
ElseIf iNewTime = 1 Then
    sNewTimeBlock = "minuut"
Else
    sNewTimeBlock = "minuten"
End If
```

We openen in bovenstaand blok een recordset, en zetten de velden die we nodig hebben in onze variabelen. Daarbij gebruiken we resp. de formules Minute en Second, omdat we in de tabel Datum/Tijd velden gebruiken. En we gaan rekenen met ofwel minuten, ofwel met seconden. Om de tekst zintechnisch straks correct te maken, vullen we ook een string met enkelvoudige of meervoudige omschrijvingen.

```
myTime = TimeSerial(0, iTime + iNewTime, 0)
myNewStartTime = Format(Now() + myTime, "hh:mm")
```

```

    strText = "Deze database gaat over " & iTime & " " & sTimeBlock
    & " automatisch dicht voor onderhoud." & vbCrLf

strText = strText & "Sluit de database zo snel mogelijk om gegevensverlies
te voorkomen."

    & vbCrLf & vbCrLf

    strText = strText & "Je kunt de database over " & iNewTime & " "
    & sNewTimeBlock & " weer openen." & vbCrLf & vbCrLf

    strText = strText & "Dat zal zijn rond " & myNewStartTime & "."

Me.lblWarning.Caption = strText

Detail.BackColor = vbYellow

Me.lblWarning.ForeColor = vbRed

End Sub

```

Het tweede blok doet de berekeningen, waarbij MyTime de twee tijden optelt, en MyNewStartTime de tijd berekent waarop de db plat gaat.

Uiteindelijk wordt de tekst in het label gezet, en worden de kleuren vastgelegd.

De Timer() functie verandert de kleuren van het formulier, zodat de gebruiker er visueel aan herinnerd wordt dat er iets staat te gebeuren.

```

Private Sub Form_Timer()

    DoCmd.Beep

    If Detail.BackColor = vbYellow Then

        Detail.BackColor = vbRed

        Me.lblWarning.ForeColor = vbYellow

    Else

        Detail.BackColor = vbYellow

        Me.lblWarning.ForeColor = vbRed

    End If

End Sub

```

Het formulier frmLogOff_Time

De tweede methode is een beetje vriendelijker voor de gebruiker, want deze controleert de gebruiker tijdens het gebruik van de db. Hierbij wordt de gebruiker uit de DB gezet als hij een ingesteld aantal minuten inactief is. Op die manier houd je het gebruik van de db redelijk goed in de hand, want als een gebruiker 15 minuten niks gedaan heeft in de db, dan mag je er wel van uit gaan dat hij/zij met iets anders bezig is. En omdat het aantal gebruikers dat tegelijk in een db werkt de prestaties van de db beïnvloeden, is het zaak om dat aantal zo laag mogelijk te houden.

Kortom: als de db 'in ruste' is, dan nemen we het recht in handen om de db af te sluiten. Daarbij kan je

dan nog kiezen voor een informatieformulier waarmee de gebruiker het roer weer kan overnemen, of je sluit alles gewoon automatisch af.

Je zou de procedure nog kunnen uitbreiden met een check op het aantal ingelogde gebruikers, en de toegang tot de db blokkeren als je vindt dat het maximum aan gebruikers ondertussen actief is. De verwijderde gebruiker moet dan wachten tot er weer een plek 'vrij komt'.

Zo ver gaan we hier niet, wij kijken alleen naar de procedure waarmee we een gebruiker uit de db kunnen krijgen na verlopen tijd.

Zoals bij het vorige formulier, gebeurt alles weer bij de gebeurtenis <Timer> van het formulier. Het *Timerinterval* geven we nu de waarde 1000, wat precies één seconde is. Het formulier werkt daarmee dus als een tikkende klok. De gebeurtenis begint weer met het declareren van een aantal variabelen die we nodig hebben.

```
Private Sub Form_Timer()
Const IDLEMINUTES = 1
Static PrevControlName As String
Static PrevFormName As String
Static ExpiredTime
Dim ActiveFormName As String
Dim ActiveControlName As String
Dim ExpiredMinutes As Variant, ExpiredSeconds As Variant
Dim db As DAO.Database
Dim snp As DAO.Recordset
Dim IDLEMINUTES As Integer
```

We gebruiken weer een aantal variabelen *Static*, omdat ze gedurende de uitvoer van de procedure niet mogen veranderen. De controle van de activiteit doen we op basis van formulieren; we gaan er vanuit dat de gebruikers alleen met formulieren mogen werken om data te bewerken, en dat ze dus niet rechtstreeks in tabellen kunnen komen. Dat is niet alleen voor de gebruikers beter, maar ook voor het beheer. We kijken dus continue of er wisseling van formulieren plaats vindt, want dat betekent doorgaans dat de gebruiker actief is.

Zoals beloofd halen we de gegevens nu rechtstreeks uit de tabel, en niet uit de Recordbron van het formulier. Dat hoeft ook niet, want op het formulier doen we er verder niks mee.

```
Set snp = CurrentDb.OpenRecordset("Settings", dbOpenSnapshot)
IDLEMINUTES = Minute(snp!LogOffTime)
```

De hele procedure draait om het vergelijken van twee formulieren: het actieve formulier, en het formulier dat de tijd registreert. De naam van het actieve formulier lezen we uit met de opdracht `ActiveFormName = Screen.ActiveForm.Name`. Hetzelfde doen we voor het control op het actieve formulier, want dat geeft ook een indicatie of er activiteit is of niet.

```
On Error Resume Next
' Get the active form and control name.
```



```

ActiveFormName = Screen.ActiveForm.Name

If Err Then

    ActiveFormName = "No Active Form"

    Err = 0

End If

ActiveControlName = Screen.ActiveControl.Name

If Err Then

    ActiveControlName = "No Active Control"

    Err = 0

End If

```

Het volgende stuk bestaat uit het controleren of er overschakeld wordt naar een ander formulier of niet. Of een ander object op het actieve formulier; voor de verwerking is dat hetzelfde. Daarbij moeten we rekening houden met het eerste gebruik van het formulier, want in dat geval is er geen vorig formulier.

Als de code voor de eerste keer draait, of er een wijziging heeft plaatsgevonden, dan moet de verstreken tijd worden genuld; die krijgt derhalve de waarde 0. Tevens wijzen we de formuliernaam en control naam toe aan de variabelen PrevFormName en PrevControlName .

In de overige gevallen wordt de verstreken tijd (ExpiredTime) verhoogd met de ingestelde Timer tijd (1 seconde).

```

If (PrevControlName = "") Or (PrevFormName = "") _
Or (ActiveFormName <> PrevFormName) Or (ActiveControlName <>
PrevControlName) Then

    PrevControlName = ActiveControlName

    PrevFormName = ActiveFormName

    ExpiredTime = 0

Else

    ExpiredTime = ExpiredTime + Me.TimerInterval

End If

```

Daarna controleren we de verstreken tijd, want uiteindelijk is het wel de bedoeling dat we een actie uitvoeren als de ingestelde tijd is verstreken.

```

ExpiredMinutes = (ExpiredTime / 1000) / 60

ExpiredSeconds = (ExpiredTime / 1000) / 60 * 60

Me.lblIdleTime.Caption = TimeSerial(0, Minute(ExpiredSeconds),
ExpiredSeconds)

Me.lblMinutes.Caption = Int(ExpiredMinutes)

```

```
Me.Repaint
```

De computertijd moet nog wel worden omgezet naar een herkenbare tijdseenheid. Dat gebeurt met de formules $\text{ExpiredMinutes} = (\text{ExpiredTime} / 1000) / 60$ en $\text{ExpiredSeconds} = (\text{ExpiredTime} / 1000) / 60 * 60$.

```
If ExpiredMinutes >= IDLEMINUTES Then
    ExpiredTime = 0
    IdleTimeDetected ExpiredMinutes
End If
End Sub
```

Als de berekende tijd (in minuten) groter is dan het aantal ingestelde minuten, moet het formulier worden gesloten. Die actie wordt uitgevoerd door de procedure `IdleTimeDetected`.

```
Function IdleTimeDetected(ExpiredMinutes)
Dim Msg As String
Dim q As Variant
    Msg = "De gebruiker " & VBA. Environ("Username") & " is gedurende " _
        & ExpiredMinutes & " minuten niet meer actief geweest."
    If q = vbYes Then Application.Quit (acQuitSaveAll)
End Function
```

In de functie wordt een `Msgbox` opgebouwd om de gebruiker een keuze te laten maken. Normaliter gebruik je dat niet, want de bedoeling is om de gebruiker die keus niet te geven. De functie hoeft dan alleen de opdracht `Application.Quit (acQuitSaveAll)` uit te voeren. De `Msgbox` zit er alleen in om de functie te kunnen testen.

Het formulier werkt het beste als je het tijdens het openen van de db inlaadt. De gebruiker hoeft het verder niet te zien, de code wordt op de achtergrond uitgevoerd. Maak dus ook een procedure die bij het starten van de db wordt uitgevoerd, en die het formulier laadt in Hidden mode. Bijvoorbeeld:

```
Public Function LaadTimerForm()
    DoCmd.OpenForm "frmLogOff_Time", acNormal, , , , acHidden
End Function
```

Deze procedure start je dan bijvoorbeeld met de macro `AutoExec`, zodat hij bij het starten van de db wordt uitgevoerd.

Samenvatting

In dit hoofdstuk heb je twee manieren gezien om gebruikers uit de database te krijgen. Eén methode gebruikt een door de beheerder ingestelde tijd en controleveld, en de tweede gebruikt een timer die de verstreken tijd meet. Welke methode je gebruikt, hangt een beetje af van de insteek van de database. Je kunt één methode gebruiken, al mag je natuurlijk ook beide methodes implementeren. Het zijn

tenslotte verschillende technieken voor verschillende doeleindes.

Als laatste nog een tip:

Als je formulieren hebt met een ingestelde Timerwaarde, dan zal Access bij elke cyclus van die tijd de procedure willen uitvoeren. Ongeacht of je op dat moment code aan het typen bent of niet. Een onverwacht, vervelend bij-effect daarvan is dat de cursor in de regel die je aan het bewerken bent verspringt. Je raakt dan bijvoorbeeld spaties kwijt, of een gereserveerd woord is ineens weg. Je hebt dus heel erg weinig te zeggen over je eigen cursor. Zet, als je dat overkomt, dan de timerwaarde op 0, en je bent van het probleem af. Is je code klaar om te testen, dan stel je de waarde weer in.

Helpmij.nl