



Access voor beginners - Hoofdstuk 16

Handleiding van Helpmij.nl

Auteur: OctaFish

December 2012

“ Dé grootste en gratis computerhelpdesk van Nederland ”

Deze maand wat handige functies die universeel inzetbaar zijn. Iedereen die programmeert heeft wel eens behoefte aan functies die je snel kunt kopiëren uit een ander bestandje, en die an meteen zonder problemen werken. Wat we in de jaren '90 OOP genoemd zouden hebben. (OOP = **O**bject **O**rientated **P**rogramming). Je zou het nu eerder "Code snippets" noemen, denk ik. Een paar snippets die ik zelf regelmatig gebruik, zal ik hier laten zien, en uitleggen.

Datumfuncties

Om te beginnen een aantal functies waarmee je snel een specifieke datum kunt berekenen. Denk bijvoorbeeld aan een standaardrapport waarmee je de omzet van de vorige week verstuurt. Zo'n week kun je natuurlijk met de hand selecteren door twee datum parameters in te voeren, maar waarom zou je dat niet gelijk goed inrichten, als je toch altijd een rapport wilt van de vorige week, van maandag tot en met vrijdag? Daarvoor kun je de volgende twee functies gebruiken als criterium.

Functies voor de huidige week

Wil je een selectie maken van de huidige week, dan kan dat met de onderstaande functies:

```
Function Begin_Week()
    Begin_Week = Date - WeekDay(Date, vbMonday) + 1
End Function
```

De bovenstaande functie berekent de maandag van de huidige week; de onderstaande doet dat voor het einde van de week.

```
Function Eind_Week()
    Eind_Week = Date - WeekDay(Date, vbMonday) + 5
End Function
```

De berekening is niet zo heel ingewikkeld; Je neemt de huidige datum, en trekt daar de huidige werkdag van af. Dat bepaal je met de functie *Weekday*. Met het argument *vbMonday* beschouwt *Weekday* de maandag als de eerste dag van de week. Zou je dat niet ingeven, dan begint *Weekday* bij zondag, en klopt de berekening niet meer. Als het dus een donderdag is, dan levert *Weekday* de waarde 4 op. Ga je op de kalender 4 dagen terug, dan kom je op zondag uit, dus je moet dan weer één dag toevoegen.

Functies voor de vorige week

Hier een vergelijkbare constructie voor twee functies die de voorgaande week bepalen:

```
Function Begin_VorigeWeek()
    Begin_VorigeWeek = Date - (WeekDay(Date, vbMonday) + 6)
End Function
```

De bovenstaande functie berekent de maandag van de vorige week, en onderstaande functie berekent weer de bijbehorende vrijdag.

```
Function Eind_VorigeWeek()
    Eind_VorigeWeek = Date - (WeekDay(Date, vbMonday) + 2)
End Function
```

De berekening is niet zo heel ingewikkeld; < Begin_VorigeWeek > je berekent weer de getalwaarde van de huidige dag met *Weekday*, telt daar 6 dagen bij op en trekt het geheel af van de huidige datum.

Toepassing:

In een query kun je een filter maken op een datumveld (bijvoorbeeld *OfferteDatum*) en dat ziet er dan zo uit:

Between Begin_VorigeWeek() AND Eind_VorigeWeek()

En daarmee is je query dus altijd voorzien van het juiste datumfilter!

Funcities voor de flexibele datums

Met een kleine aanpassing kun je de start- en einddatum ook flexibel maken. Hierbij hoef je alleen maar een parameter toe te voegen aan de functie en met deze datum te rekenen.

```
Function BeginVanWeek(StartDatum As Date)
    BeginVanWeek = StartDatum - WeekDay(StartDatum, vbMonday) + 1
End Function
En voor de einddatum:
Function EindVanWeek(EindDatum As Date)
    EindVanWeek = EindDatum - WeekDay(EindDatum, vbMonday) + 7
End Function
```

In deze functies wordt om een (datum)veld gevraagd. De waarde daaruit kan uit een parameter komen die bij het starten van een query wordt gevraagd, of uit een formulier waarop je een datum aanklikt.

Between BeginVanWeek([Typ de datum]) AND EindVanWeek([Typ de datum])

Je gebruikt dezelfde parameter als je één week wilt zien, en verschillende parameters als je een grotere reeks wilt hebben.

Leeftijd berekenen

Een vraag die regelmatig voorbijkomt is: hoe kan ik het beste de leeftijd van een persoon uitrekenen? Het antwoord is dan natuurlijk: met een functie!

```
Function Leeftijd(varBegindatum As Variant, Optional Peildatum As Date) As Integer
*****
' DOEL:
' Leeftijd in jaren berekenen, vanaf de opgegeven datum.
' INVOERPARAMETERS:
' varBegindatum: De begindatum (bijvoorbeeld een geboortedatum).
' Peildatum: De einddatum (optioneel; bij leeglaten huidige datum).
' RESULTAAT
' De leeftijd in jaren.
*****
Dim varAge As Variant

If IsNull(varBegindatum) Then Leeftijd = 0: Exit Function
varBegindatum = CDate(varBegindatum)
If CLng(PeilDatum) = 0 Then PeilDatum = Date
varAge = DateDiff("yyyy", varBegindatum, PeilDatum)
If PeilDatum < DateSerial(Year(PeilDatum), Month(varBegindatum), Day(varBegindatum)) Then
    varAge = varAge - 1
End If
Leeftijd = CInt(varAge)
End Function
```

Weer een kleine toelichting:

Je kunt de natuurlijke leeftijd van een persoon (of iets anders natuurlijk) berekenen (hoe oud is de persoon nu) of t.o.v. een bepaalde peildatum (Hoe oud was de persoon op 1 september bijvoorbeeld). In het laatste geval heb je twee datums nodig: de geboortedatum en de peildatum. De geboortedatum

wordt als Variant gedefinieerd, zodat je makkelijker kunt controleren of het veld niet leeg is gelaten. De peildatum is wel een datumveld. Daarop wordt in de functie een check gedaan.

Met CDate wordt de variabele varBeginDatum geconverteerd naar een datum. Als het de parameter Peildatum leeg is, krijgt die de huidige datum. Je hebt dus in de functie DateDiff, die het verschil in jaren tussen de twee berekent, namelijk de datums in varBeginDatum en Peildatum. Daar komt een getal uit. Dat is niet in alle gevallen correct; een persoon die op 15-12-90 is geboren, is op 14-12-90 nog niet jarig geweest. DateDiff zal de leeftijd op basis van het aftrekken van de jaartallen als 22 uitrekenen. Maar er moet dus nog (tijdelijk) een jaar vanaf worden getrokken. Dat gebeurt in de IF constructie.

Berekenen van weeknummers

Een andere populaire vraag is, hoe je een weeknummer kunt berekenen. Dat doen heel veel mensen met de functie Format (voorbeeldje: Format([Geboortedatum];"ww";2;2)) maar deze manier van berekenen levert nog wel eens een fout uitkomst op. Beter is het om een weeknummer te berekenen op basis van de ISO norm. En dat gaat dan zo:

```
Public Function WeekNum(ZoekDatum As Date, Optional WhichFormat As Variant) As Integer
```

```
'Versie laatst bijgewerkt: 2-4-2008
```

```
Dim PreviousYearStart As Date, ThisYearStart As Date, NextYearStart As Date
```

```
Dim ThisYear As Integer, YearNum As Integer
```

```
ThisYear = Year(ZoekDatum)
```

```
ThisYearStart = YearStart(ThisYear)
```

```
PreviousYearStart = YearStart(ThisYear - 1)
```

```
NextYearStart = YearStart(ThisYear + 1)
```

```
Select Case ZoekDatum
```

```
Case Is >= NextYearStart
```

```
WeekNum = (ZoekDatum - NextYearStart) 7 + 1
```

```
YearNum = Year(AnyDate) + 1
```

```
Case Is < ThisYearStart
```

```
WeekNum = (ZoekDatum - PreviousYearStart) 7 + 1
```

```
YearNum = Year(ZoekDatum) - 1
```

```
Case Else
```

```
WeekNum = (ZoekDatum - ThisYearStart) 7 + 1
```

```
YearNum = Year(ZoekDatum)
```

```
End Select
```

```
If IsMissing(WhichFormat) Then
```

```
Exit Function
```

```
End If
```

```
If WhichFormat = 2 Then WeekNum = CInt(Format(Right(YearNum, 2), "00") & Format(WeekNum, "00"))
```

```
End Function
```

Deze functie gebruikt een hulpfunctie om de startdatum te berekenen. Die ziet er zo uit:

```
Private Function YearStart(WhichYear As Integer) As Date
```

```
'Versie laatst bijgewerkt: 2-4-2008
```

```
Dim WeekDay As Integer, NewYear As Date
```

```
NewYear = DateSerial(WhichYear, 1, 1)
```

```
WeekDay = (NewYear - 2) Mod 7
```

```
If WeekDay < 4 Then
```

```
YearStart = NewYear - WeekDay
```

```
Else
```

```

YearStart = NewYear - WeekDay + 7
End If
End Function

```

Hierbij wordt de startdatum bepaald aan de hand van de eerste dag van het jaar waarbij er minstens 4 dagen in de week zitten. Wat meestal de gewenste startdatum oplevert.

Om bij de hoofdfunctie terug te keren: die begint met het vastleggen van een aantal variabelen. Vervolgens wordt met een SELECT CASE het weeknummer berekend, door de begin- en einddatum van elkaar af te trekken en door 7 te delen. Daarbij wordt een grappige truc gebruikt, die waarschijnlijk niet gelijk opvalt. Kijk maar eens naar deze twee formules:

Expr4: (Date()-[Geboortedatum]) / 7

Expr5: (Date()-[Geboortedatum]) \ 7

In beide gevallen wordt de leeftijd in weken berekend op basis van geboortedatum en huidige datum. Alleen zijn de uitkomsten verschillend:

Expr4: 1154,7142857

Expr5: 1154

De eerste deling rekent op de normale manier, waarbij een rest in de uitkomst kan zitten. Als je iets door 7 deelt, is die kans al heel groot tenslotte.

De tweede variant laat alleen het gehele getal zien; het lijkt er op alsof je de hele berekening ook nog eens in de functie CInt hebt genest. Maar dat is dus niet zo: je bereikt dit effect al door een deling te maken met de backslash (\) i.p.v. te delen door de slash (/). En dat maakt de berekening natuurlijk een stuk korter! Gebruik het (backslash)deelteken dus niet per ongeluk verkeerd, want dan klopt je berekening vermoedelijk niet meer.

Omdat een weeknummer nooit met 1 begint, wordt bij de uitkomst nog de waarde 1 opgeteld.

Deze werkwijze is dus in beginsel wel ingewikkelder dan de Format optie, maar levert wel altijd het juiste resultaat op. Ik zou je dus aan willen bevelen om weeknummers altijd met deze functie te berekenen!

Functie als Rijbron voor een keuzelijst

We zijn gewend om keuzelijsten te maken die zijn gebaseerd op een tabel, of een lijst met waarden. Dat vind je ook in de Help functie terug; als je ziet welke mogelijkheden je hebt voor een keuzelijst, lees je dit:

De eigenschap **RowSourceType** kent de volgende instellingen.

Instelling	Beschrijving
Tabel/query	(Standaard) De gegevens komen uit de tabel, query of SQL-instructie SQL-reeks/instructie: een expressie waarmee een SQL-opdracht wordt gedefinieerd, bijvoorbeeld SELECT, UPDATE of DELETE, en die componenten bevat als WHERE en ORDER BY. SQL-reeksen/instructies worden doorgaans toegepast in query's en statistische functies.) die wordt aangegeven met de instelling van RowSource .
Lijst met waarden	De gegevens bestaan uit de lijst met elementen die wordt aangegeven door de instelling van RowSource .
Lijst met velden	De gegevens bestaan uit de lijst met veldnamen uit een tabel, query of SQL-instructie die wordt aangegeven door de instelling van RowSource .

En dat is dan ook wat er in 99% van de gevallen gebeurt. Meestal baseer je een keuzelijst op een tabel, soms op een lijst met waarden en in een uitzonderlijk geval gebruik je Lijst met velden.

Maar je kunt ook een functie gebruiken, als je even doorleest in de Help!

Opmerking

U kunt de eigenschap **RowSourceType** ook instellen met een door de gebruiker gedefinieerde functie (door gebruiker gedefinieerde functie: een query die parameters nodig heeft en een resultaat geeft, zoals een opgeslagen procedure. Typen: scalair (meerdere instructies; geeft één waarde als resultaat), in line (één instructie; een tabelwaarde die kan worden bijgewerkt) en tabel (meerdere instructies; tabelwaarde).). De functienaam wordt ingevoerd zonder voorafgaand gelijkteken (=) en zonder volgend paar haakjes. U moet [bepaalde codeargumenten](#) opgeven voor de functie om aan te geven hoe het besturingselement moet worden gevuld.

De moeilijkheidsgraad hierbij ligt in de paar woorden *bepaalde codeargumenten*. Een keuzelijst moet aan nogal wat voorwaarden voldoen om hem werkend te krijgen. Omdat we een uitdaging niet uit de weg gaan, maken we een functie waarmee we een dynamische lijst met datums gaan maken. Stel dat je een rapport hebt gemaakt waarin je een roosterplanning laat zien voor een bepaalde periode. Zodat je een goed inzicht hebt in de personele bezetting bijvoorbeeld. Dan zul je de ene keer het rooster voor de komende week willen zien, en een volgende keer misschien een rooster voor de komende maand. Misschien wil je het rooster wel afdrucken voor een periode van 3-5 weken in de toekomst.

Nu kun je zo'n rapport prima baseren op twee datumvelden die je op een formulier gebruikt om middels een kalender de start- en einddatum te kiezen. Maar zo'n rapport zul je willen baseren op een vaste tijdreeks: altijd beginnen op de eerste (werk)dag van de eerste week, en lopen tot de laatste (werk)dag van de laatste week. Dus waarom niet twee keuzelijsten maken, waarin je gelijk de juiste dagen selecteert?

Probleem is dan, dat je in die keuzelijst niet alle weken van het jaar wilt opnemen. Daar zou de keuzelijst namelijk behoorlijk lang van worden. Bovendien: bij een roosterplanning gaat het om datums die in de toekomst liggen, dus startdatums in het verleden zijn niet interessant, maar daar moet dan wel doorheen gescrolld worden. In die situatie kan een functie heel goed werken.

Eisen voor de functie

Zoals hierboven al gemeld, moet een functie aan een aantal *codeargumenten* voldoen wil hij werken. Als je in de Help op Codeargumenten klikt, zie je welke argumenten dat zijn.

Function *functionname* (**fld As Control**, **id As Variant**, **row As Variant**, **col As Variant**, **code As Variant**) As Variant

Gelukkig geeft Microsoft een prima voorbeeldje, dat we kunnen gebruiken voor de functie. Die krijgt als naam: LijstMaandagenOp.

Function LijstMaandagenOp(vld As Control, Id As Variant, rij As Variant, kol As Variant, Code As Variant) As Variant
Dim intOffset As Integer

De functie declareert een aantal argumenten die gebruikt worden, zoals de control die we gaan gebruiken (de keuzelijst) en het variabeletype dat gebruikt wordt als resultaat voor de functie (Variant). Een keuzelijst heeft verschillende aspecten, die met het argument Code worden uitgelezen en ingesteld. Niet alle elementen hoeven te worden ingesteld, dus de gewenste aspecten worden met een Select Case uitgelezen. Er zijn in totaal 9 constanten die doorlopen worden, beginnend bij 0 (acLBInitialize) tot 9 (acLBEnd).

Select Case Code

```
Case acLBInitialize      ' Initialiseren.
    LijstMaandagenOp = True
```

Met *acLBInitialize* wordt de keuzelijst geactiveerd

```
Case acLBOpen          ' Openen.
    LijstMaandagenOp = Timer    ' Unieke id.
```

De functie krijgt een unieke ID. Door daar de functie *Timer* voor te gebruiken, ben je verzekerd van een uniek getal.

```
Case acLBGetRowCount    ' Aantal rijen ophalen.
    If vld.ListRows > 16 Then
        LijstMaandagenOp = 16
    Else
        LijstMaandagenOp = vld.ListRows
    End If
```

Het volgend aspect wordt gebruikt om het aantal elementen van de keuzelijst in te stellen. Hier kun je een vaste waarde ingeven, maar je kunt de lijst ook dynamisch maken, door het aantal waarden gelijk te maken aan het aantal rijen dat je ingeeft bij de eigenschap *<Aantal rijen>* van de keuzelijst. Op die manier heb je altijd voldoende waarden in de lijst staan. Om te voorkomen dat er teveel waarden worden aangemaakt, kun je met een *IF...End IF* het maximale aantal rijen vastleggen.

```
Case acLBGetColumnCount ' Aantal kolommen ophalen.
    LijstMaandagenOp = 1
```

Je kunt ook het aantal kolommen instellen, net zoals je dat bij de keuzelijst zelf doet. Gebruik je meer dan één kolom, dan moet je de gegenereerde waarden scheiden met een komma. In het voorbeeld maken we voor het gemak een keuzelijst met één waarde.

```
Case acLBGetColumnWidth ' Kolombreedte ophalen.
    LijstMaandagenOp = -1 ' Standaardbreedte gebruiken.
```

Met deze instelling bepaal je de breedte van de kolom. De standaardwaarde voldoet hier, want die kun je instellen bij de *Eigenschappen* van de keuzelijst.

```
Case acLBGetValue      ' Gegevens ophalen.
    intOffset = Abs((9 - WeekDay(Date)) Mod 7)
    LijstMaandagenOp = Format((Date - 7) + intOffset + 7 * rij, "d mmmm")
End Select
End Function
```

Het belangrijkste, want het aanmaken van de noodzakelijke waarden van de keuzelijst, gebeurt bij de constante *acLBGetValue*. De code bestaat uit twee regels:

```
intOffset = Abs((9 - WeekDay(Date)) Mod 7)
```

en

```
LijstMaandagenOp = Format((Date + intOffset - 7) + 7 * rij, "d mmmm")
```

De keuzelijst moet een lijst met maandagen genereren. We moeten dus eerst een manier vinden om een uit te rekenen hoeveel dagen er liggen tussen de huidige datum en de volgende maandag. Dat doen we door het restbedrag van de *Weekdag* van de huidige datum gedeeld door 7 te berekenen, en dat getal af te trekken van het getal 9. De uitkomst zetten we in de variabele *intOffset*. In de volgende regel wordt bij de huidige datum één week de variabele *intOffset* opgeteld (levert de

eerstvolgende maandag op) en de waarde 7 afgetrokken om met de maandag van de huidige week te beginnen. Omdat we voor elke rij in de keuzelijst een volgende week nodig hebben, wordt daarna het rijnummer (in de variabele rij) vermenigvuldigd met 7 opgeteld bij de nieuwe datum. De eerste rij heeft de waarde 0, dus de lijst begint met de eerst berekende datum.

De complete functie ziet er dus zo uit:

```
Function LijstMaandagenOp(vld As Control, Id As Variant, rij As Variant, kol As Variant, Code As Variant) As Variant
Dim intOffset As Integer
Select Case Code
    Case acLBInitialize      ' Initialiseren.
        LijstMaandagenOp = True
    Case acLBOpen           ' Openen.
        LijstMaandagenOp = Timer ' Unieke id.
    Case acLBGetRowCount    ' Aantal rijen ophalen.
        If vld.ListRows > 16 Then
            LijstMaandagenOp = 16
        Else
            LijstMaandagenOp = vld.ListRows
        End If
    Case acLBGetColumnCount ' Aantal kolommen ophalen.
        LijstMaandagenOp = 1
    Case acLBGetColumnWidth ' Kolombreedte ophalen.
        LijstMaandagenOp = -1 ' Standaardbreedte gebruiken.
    Case acLBGetValue       ' Gegevens ophalen.
        intOffset = Abs((9 - WeekDay(Date)) Mod 7)
        LijstMaandagenOp = Format((Date + intOffset - 7) + 7 * rij, "d mmmm")
End Select
End Function
```

Opdracht

Maak een keuzelijst die op basis van het aantal rijen in de keuzelijst aflopende weeknummers laat zien. Gebruik hierbij de functie *WeekNum* die we eerder in het hoofdstuk hebben gemaakt.

Eigenschappen van een function procedure

Ter informatie nog uit de Help functie van Access de beschrijvingen van de verschillende functie constanten.

De **Function**-procedure kent de volgende vijf verplichte argumenten.

Argument Beschrijving

- vld* Een variabele voor een besturingselement die verwijst naar de keuzelijst (keuzelijst: een besturingselement dat bestaat uit een lijst met keuzen. Een keuzelijst bestaat uit een lijst en een optionele label.) of keuzelijst met invoervak (keuzelijst met invoervak: een besturingselement dat in een formulier wordt gebruikt en de functionaliteit van een keuzelijst en een tekstvak combineert. In een keuzelijst met invoervak kunt u een waarde typen of kunt u klikken op het besturingselement en een item in de lijst selecteren.) die wordt gevuld.
- id* Een unieke waarde die het besturingselement aanduidt dat wordt gevuld. Dit is handig als u dezelfde door de gebruiker gedefinieerde functie voor meerdere keuzelijsten of keuzelijsten met invoervakken wilt gebruiken en u onderscheid wilt kunnen maken tussen de verschillende besturingselementen. (In het voorbeeld wordt deze variabele ingesteld op de waarde van de functie **Timer**.)
- rij* De rij die wordt gevuld (gebaseerd op nul).
- kol* De kolom die wordt gevuld (gebaseerd op nul).
- code* Een intrinsieke constante (ingebouwde constante: een constante die wordt geleverd door

Access, VBA, ADO of DAO. Deze constanten zijn beschikbaar in het Objectenoverzicht door te klikken op globalen in elk van deze bibliotheken.) die het type gevraagde informatie aanduidt.

Opmerking

Aangezien een door de gebruiker gedefinieerde functie verschillende keren wordt aangeroepen om items in een lijst in te voegen, moeten gegevens worden opgeslagen tussen de aanroepen door. U kunt hiervoor het beste statische variabelen gebruiken.

Microsoft Access roept de door de gebruiker gedefinieerde functie herhaaldelijk op, met verschillende waarden in het argument *code* om de benodigde waarden op te geven. Het argument *code* kan de volgende intrinsieke constanten gebruiken.

Constante	Betekenis	Resultaat van de functie
acLBInitialize	Initialiseren	Niet-nul als de functie de lijst kan vullen. Anders False (0) of Null (Null: een waarde die kan worden ingevoerd in een veld of worden gebruikt in expressies en query's om ontbrekende of onbekende gegevens aan te duiden. In Visual Basic verwijst het sleutelwoord Null naar een Null-waarde. Sommige velden, zoals velden waarvoor een primaire sleutel is ingesteld, kunnen geen Null-waarden bevatten.).
acLBOpen	Openen	Id-waarde, niet-nul, als de functie de lijst kan vullen. Anders False of Null .
acLBGetRowCount	Aantal rijen	Aantal rijen in de lijst (kan nul zijn); -1 indien onbekend.
acLBGetColumnCount	Aantal kolommen	Aantal kolommen in de lijst (kan niet nul zijn); moet overeenkomen met de waarde in het eigenschappenvenster.
acLBGetColumnWidth	Kolombreedte	Breedte (in twips (twip: een maateenheid die gelijk is aan 1/20 van een punt of 1/1.440 inch. Er gaan 567 twips in een centimeter.)) van de kolom die is aangeduid in het argument <i>kol</i> , -1 voor de standaardbreedte.
acLBGetValue	Lijstinvoer	Lijstinvoer die moet worden weergegeven in de rij en de kolom aangeduid met de argumenten <i>rij</i> en <i>kol</i> .
acLBGetFormat	Notatiereeks	Notatiereeks die moet worden gebruikt om de lijstinput weer te geven in de rij en de kolom die zijn aangeduid in de argumenten <i>rij</i> en <i>kol</i> , -1 voor de standaardnotatie.
acLBEnd	Einde (altijd de waarde voor de laatste oproep voor een functie die door de gebruiker is gedefinieerd)	Niets.
acLBClose	(Niet gebruikt)	Niet gebruikt.

Microsoft Access roept de door de gebruiker gedefinieerde functie éénmaal op voor **acLBInitialize**, **acLBOpen**, **acLBGetRowCount** en **acLBGetColumnCount**. De door de gebruiker gedefinieerde functie wordt geïnitieerd, de query wordt geopend en het aantal rijen en kolommen wordt bepaald.

Microsoft Access roept de door de gebruiker gedefinieerde functie tweemaal op voor

acLBGetColumnWidth, eerst om de totale breedte van de keuzelijst of keuzelijst met invoervak te bepalen en vervolgens om de kolombreedte in te stellen.

Het aantal keren dat de door de gebruiker gedefinieerde functie wordt opgeroepen voor **acLBGetValue** en **acLBGetFormat** om lijstinput op te halen en notatiereeksen in te stellen is afhankelijk van het aantal ingevoerde elementen, het gebruik van de schuiftoetsen door de gebruiker en andere factoren.

De door de gebruiker gedefinieerde functie voor **acLBEnd** wordt opgeroepen wanneer het formulier wordt gesloten, of telkens wanneer een query wordt uitgevoerd voor de keuzelijst of keuzelijst met invoervak.

Telkens wanneer een bepaalde waarde (zoals het aantal kolommen) vereist is en als het resultaat **Null** of een andere ongeldige waarde is, wordt gestopt met het oproepen van de door de gebruiker gedefinieerde functie met die code.

Helpmij.nl