



## TreeView i Access III

**Skulle det nu være nødvendigt med en tredje udgave af treeview. Måske ikke, men jeg havde behov for at sætte et treeview op over en virksomhed og fandt efter en Netsøgning på en række ting, som ikke har været nævnt før.**

Skrevet den **04. Feb 2009** af **hnteknik** | kategorien **Databaser / Access** | ★★★★★

TreeView i Access III

21-01-06 : Demodatabasen kan downloades herfra:

<http://www.web4it.dk/download/HENTreviewdemo.zip>

Skulle det nu være nødvendigt med en tredje udgave af treeview. Måske ikke, men jeg havde behov for at sætte et treeview op over en virksomhed og fandt efter en Netsøgning på en række ting, som ikke har været nævnt før.

### 1) Vi starter med at lave en simpel database.

TblKoncern

Kid	ID	autonr.	
Knavn	koncernnavn		string
KAdr1	Koncernadresse		string
....			

TblFirma

Fid	ID	autonr	
Kid	Moderkoncern		long
Fnavn	Firmanavn (Division)		string
Fadr1	Firmaadresse		string
...			

TblAdr

Aid	ID	autonr	
Fid	Moderfirma		long
Anavn	Lokationsnavn		string
Aadr1	Lokationsadresse		string
...			

Der laves referentiel integritet med en til mange relationer mellem de tre tabeller, så Koncern->Firma->Adresse hierarkiet oprettes. Der kan selvfølgelig være en hel masse andre parametre i disse tabeller samt andre relevante tabeller knyttet på forskellige steder i træet. Men princippet er det samme.

Til brug i treeview kontrollen senere opretter vi 3 forespørgelser:

```
KonList: SELECT Tblkoncern.Knavn, Tblkoncern.kid FROM Tblkoncern ORDER BY Tblkoncern.Knavn;
```

```
DivList: SELECT Tblfirma.Fnavn, Tblfirma.Fid FROM Tblfirma WHERE ((Tblfirma.Kid)=[Koncernen]))  
ORDER BY Tblfirma.Fnavn;
```

```
AdrList: SELECT TblAdr.AAdr1 AS Adresse, TblAdr.Aid FROM TblAdr WHERE (((TblAdr.Fid)=[Divisionen]))  
ORDER BY TblAdr.AAdr1;
```

Queriene er dels for at have et ultrakort udtræk fra de bagvedliggende tabeller, som kan indeholde en række andre oplysninger men også for at begrænse resultaterne ved hjælp af parametrene [Koncernen] og [Divisionen]. Disse to parametres anvendelse kommer vi tilbage til. Hvad de hedder er ikke så afgørende i den senere brug.

Læg en række dummy data ind for at have noget at præsentere senere.

## 2) Vi opretter en formular

Der oprettes en ny form FrmTreeExplorer med:

En formular header foreløbig med en label: Treeviewdemo og en standard lukkeknop fra menuen. Fjern formular foden, den skal vi ikke bruge nu.

I detaildelen laver vi baggrunden mørkegrå - baggrundsfarve: 5987163

Husk at klikke nej til navigation, scrollbar og postvælgere. Denne form skal ikke referere direkte til nogen tabel.

Fra den udvidede værktøjskasse trækker vi en treeviewkontrol hen på detailsiden venstre side. Størrelse ca 10 x 12 cm med lidt luft omkring, så man kan se den grå farve. Af kontrol til Access 2003 bruger jeg 'Microsoft Treeview Control 6 (SP6)'. Omdøb objektet til 'ExplorerPane'

Fra den udvidede værktøjskasse trækker vi nu en ImageList Kontrol (til vores ikoner) hen på formular headeren. Her bruger jeg en kontrol med navn 'Microsoft Imagelist Control 6 (SP6)'. Placering er ikke så vigtig, da den ikke er synlig. Omdøb objektet til f.eks. MyImageList.

Højreklik på kontrollen 'ExplorerPane' og vælg Treectrl-object ->properties

Vælg her MyImageList som Imagelist og tryk 'OK'.

Højreklik på kontrollen MyImageList og vælg Imagelistctrl-objekt ->properties

Klik på fanen 'Images' og indsæt 3 ikoner for 'koncern', 'Firma' og 'Adresse' ved at klikke på 'Insert Picture'. Vælg nogle tilfældige fra systemet, hvis du ikke har nogen passende liggende.

Jeg har indsat tre gif filer 'Stjerne', 'Højhus' og 'villaer' som eksempel og givet dem deres respektive KEY og TAG : 'Koncern', 'Firma' og 'Adresse'. Tryk 'OK'.

Husk at tage friske kopier af de to kontroller, ellers kan du opleve, at det ikke fungerer, hvis du blot kopierer objekterne fra en anden formular.

## 3)Vi skal nu have vores Tree Kontrol til at virke med lidt kode behind:

Gå ind i programkoden bagved formularen og indsæt følgende kode:

```

Private Sub Form_Load()
    LoadTree
End Sub

Public Sub LoadTree()
    Dim rsAdr As DAO.Recordset, rsDiv As DAO.Recordset, rsKoncern As DAO.Recordset
    Dim qryAdr As DAO.QueryDef, qryDiv As DAO.QueryDef
    Dim theAdr As Node, theKon As Node, theDIV As Node
    Dim sProcessName As String

    ' indsæt koncerter her -----
    Set rsKoncern = CurrentDb.OpenRecordset("KonList", dbOpenForwardOnly)
    Do While Not rsKoncern.EOF
        sProcessName = "K" & rsKoncern!kid
        Set theKon = Me.ExplorerPane.Nodes.Add(, , sProcessName, rsKoncern!Knavn, "Koncern")
        theKon.Tag = "Koncern"
        ' indsæt Divison her -----
        Set qryDiv = CurrentDb.QueryDefs("DivList")
        With qryDiv
            .Parameters(0) = rsKoncern!kid
            Set rsDiv = .OpenRecordset(dbOpenForwardOnly)
            Do While Not rsDiv.EOF
                sProcessName = "D" & rsDiv!fid
                Set theDIV = Me.ExplorerPane.Nodes.Add(theKon, tvwChild, sProcessName, rsDiv!fnavn,
"Firma")
                theDIV.Tag = "Division"

                ' indsæt adresser her -----
                Set qryBygn = CurrentDb.QueryDefs("AdrList")
                With qryBygn
                    .Parameters(0) = rsDiv!fid
                    Set rsAdr = .OpenRecordset(dbOpenForwardOnly)
                    Do While Not rsAdr.EOF
                        sProcessName = "A" & rsAdr!Aid
                        Set theAdr = Me.ExplorerPane.Nodes.Add(theDIV, tvwChild, sProcessName,
rsAdr!Adresse, "Adresse")
                        theAdr.Tag = "Adresse"
                        rsAdr.MoveNext
                    Loop
                    rsAdr.Close
                    Set rsAdr = Nothing
                End With
                rsDiv.MoveNext
            Loop
            rsDiv.Close
            Set rsDiv = Nothing
        End With
        rsKoncern.MoveNext
    Loop
    rsKoncern.Close
    Set rsKoncern = Nothing
End Sub

```

## Forklaring til koden:

```
sProcessName = "K" & rsKoncern!kid
```

Alle ID'er dannes ved hjælp af autonummerering så for at sikre en unik node Key, sætter vi et bogstav K, D eller A foran - alt efter hvilken tabel, vi henter data fra. Samtidig carster vi ID til en String, som anvendes i Treeview kontrollen.

```
Set theKon = Me.ExplorerPane.Nodes.Add(, sProcessName, rsKoncern!Knavn, "Koncern")
```

theKon er koncern noden som er masternoden - hvorfor de to første felter er tomme, sProcessName er Node Key fra oven. rsKoncern!Knavn er den tekst, jeg har valgt at vise ude for noden og "Koncern" er en reference til vores GIF fil fra før.

```
theKon.Tag = "Koncern"
```

er Taggen for denne type nodes, som vi kan referere til.

Umiddelbart efter koncern noden indsættes de subnotes, som hører til denne koncern. Det gør vi ved at kalde vores query for firmaer,/divisioner, som vi dannede i starten med en parameter til det vi kaldte [Koncernen], som får tildelt den overliggende koncernnodes ID

```
.Parameters(0) = rsKoncern!kid
```

Sådan arbejdes der hierarkisk ned gennem de enkelte knudepunkter indtil hele træet er opbygget.

Når formularen nu vises skulle der gerne opbygges et lille hierarkisk træ. Klik på koncernikonet (stjerne i mit tilfælde) og udfold træet ved gentagen klikken på de enkelte noder.

Sæt endnu en knap 'Udfold træ' med navnet 'BtnOpenTree' i headeren af formularen og tilføj følgende kode:

```
Private Sub BtnOpenTree_Click()  
Dim Node As Node  
    'Expand all nodes  
    For Each Node In ExplorerPane.Nodes  
        Node.Expanded = True  
    Next  
End Sub
```

Nu kan hele træet udfoldes blot ved et klik.

En tilsvarende knap 'Sammenfold træ' med navnet 'BtnCloseTree' pakker det hele sammen:

```
Private Sub BtnCloseTree_Click()
```

```

Dim Node As Node
  'Collaps all nodes
  For Each Node In ExplorerPane.Nodes
    Node.Expanded = False
  Next
End Sub

```

#### 4) Tilføj forskellige subforms som panes:

Det kan være meget sjovt at kunne traversere et treeview, men vi vil gerne have vist nogle flere data end dem, der er vist i træet. Det gør vi i subforms som vi placerer i højre side og her kalder panes:

#### AdrPane, Divpane, KoncernPane

De tre formularer får fjernet navigation, postvælgere og scrollbars og refereres til de respektive tabeller, tbladr, tblfirma og tblkoncern. De få attributter, vi definerede i de enkelte tabeller, sættes ind på formen ( Du er velkommen til at sætte flere ind), hvor vi skjuler Id'et, da der kun skal refereres til det.

Det er en god ide, at give de tre forms samme størrelse og afpasses explorerpanes størrelse. De trækkes ind på vores hovedformular - først adrpanen - fjern labelen og klik nej i vis. Dernæst trækkes DivPane ind. Denne får også nej i vis og får fjernet label. KoncernPane trækkes også ind og får fjernet label. De tre forms alignes over hinanden så de har samme position og størrelse, så det ikke hopper, når vi skifter mellem dem om lidt.

#### 5) Tilføj code behind til hovedformularen for at styre panes:

Som det er nu, sker der ikke så meget - Tilføj først en label i headeren af masterformen - lav den lang nok og kald den LblNode med en dummy tekst 'XXX'.

#### 6) Tilføj så denne kode til masterformen:

```

Private Sub ClearHighlight()
Dim n As Node
  For Each n In ExplorerPane.Nodes
    If Not n.Selected Then
      n.BackColor = vbWindowBackground
      n.ForeColor = vbWindowText
    End If
  Next
End Sub

Private Sub ExplorerPane_NodeClick(ByVal Node As Object)
Dim thePane As Form, Breadcrumbs As String
Dim nodItem As Node, nodRoot As Node

Me.ExplorerPane.SetFocus ' måtte lige sætte focus på noget herovre når der klikkes fra søgeform
ClearHighlight ' rens lige alle andre end den selectede
Select Case Node.Tag '
  Case "Koncern"
    Me.Adressepane.Visible = False
    Me.DivisionPane.Visible = False

```

```
Set thePane = Me.KoncernPane.Form
thePane.Filter = "KID = " & Mid(Node.Key, 2)
thePane.FilterOn = True
Me.KoncernPane.Visible = True
```

```
Case "Division"
```

```
Me.Adressepane.Visible = False
Me.KoncernPane.Visible = False
```

```
Set thePane = Me.DivisionPane.Form
thePane.Filter = "FID = " & Mid(Node.Key, 2)
thePane.FilterOn = True
Me.DivisionPane.Visible = True
```

```
Case "Adresse"
```

```
Me.DivisionPane.Visible = False
Me.KoncernPane.Visible = False
```

```
Set thePane = Me.Adressepane.Form
thePane.Filter = "AID = " & Mid(Node.Key, 2)
thePane.FilterOn = True
Me.Adressepane.Visible = True
```

```
End Select
```

```
' LIDT BREADCRUMB CODING :-) -----
```

```
Breadcrumbs = ExplorerPane.SelectedItem
Set nodItem = Node.Parent
Set nodRoot = Node.Root
```

```
If nodItem Is Nothing Then
```

```
Else
```

```
While Not nodItem = nodRoot
    Breadcrumbs = nodItem.Text & "->" & Breadcrumbs
    Set nodItem = nodItem.Parent
Wend
```

```
Breadcrumbs = nodRoot.Text & "->" & Breadcrumbs
```

```
End If
```

```
Me.LblNode.Caption = Breadcrumbs
```

```
End Sub
```

Forklaring til koden:

ClearHighlight er en generel sub, som genskaber farven på tekst og baggrund til default på alle de nodes, som ikke er selected. Hvorfor nu det. Jo senere skal vi selecte ved et kald fra en anden form, og da det ikke direkte er et klik med en mus, opdateres farverne ikke automatisk. Derfor denne lille udrensning.

Når vi kalder fra en anden form, kan vores ExplorerPane komme lidt ud af fokus, så derfor må vi først sætte fokus:

```
Me.ExplorerPane.SetFocus
```

```
Select Case Node.Tag '  
Case "Koncern"
```

Da vi indlæste vores tree, gav vi de enkelte typer nodes et tag i.e. 'Koncern'. Når der klikkes på en node, så handler vi efter hvilken tag noden har.

```
Me.Adressepane.Visible = False  
Me.DivisionPane.Visible = False
```

Vi sikrer først, at de ikke-aktuelle Panes ikke er visible.

Dernæst vælges den ønskede Pane og data filtreres, hvorefter vi gør panen visibel:

```
Set thePane = Me.KoncernPane.Form  
thePane.Filter = "KID = " & Mid(Node.Key, 2)  
thePane.FilterOn = True  
Me.KoncernPane.Visible = True
```

Bemærk at vi lige skræller bogstavet - i dette tilfælde K af Keyen inden det anvendes som ID i filtret.

Vi vises nu den rigtige Pane med data for den node, som vi klikkede på.

Sluttelig skal vi lige lave en Breadcrumb for at vise, hvor vi er i hierarkiet:

```
Breadcrumbs = ExplorerPane.SelectedItem  
Set nodItem = Node.Parent  
Set nodRoot = Node.Root  
  
If nodItem Is Nothing Then  
Else  
While Not nodItem = nodRoot  
Breadcrumbs = nodItem.Text & "->" & Breadcrumbs  
Set nodItem = nodItem.Parent  
Wend  
Breadcrumbs = nodRoot.Text & "->" & Breadcrumbs  
End If  
Me.LblNode.Caption = Breadcrumbs
```

Vi finder først indholdet I selecteditem for ExplorerPane og arbejder os op via Parent nodes til vi rammer rodnoden, som er vores koncern. Dernæst vises Breadcrumbs som en Caption i vores label fra før.

Nu kan vi bevæge os rundt i træet og se indholdet af de forskellige tabellers records.

## 7) Hvordan jumpes der til en bestemt node i træet

Indledningsvis har jeg indikeret, hvordan man kan jumpe til en bestemt node i træet fra en anden form, f.eks. en søge form. Det vil fylde for meget her, hvis jeg skulle komme ind på hele koden bag et søgejump og viser blot den kode, der hører vores explorer.

For at kunne lave et jump, skal vi have et kald til Springtil\_Aktiv\_Click fra et par form actions (Form POPUP ved højreklik i træet må vi komme ind på en anden gang):

```
Private Sub Form_Activate()  
    Springtil_Aktiv_Click  
End Sub  
  
Private Sub Form_Load()  
    LoadTree  
    Springtil_Aktiv_Click  
    'Gemmer den til andre tider ;-)  
    'Form_popup  
End Sub
```

Sub koden ser sådan ud:

```
Private Sub Springtil_Aktiv_Click()  
    Dim n As Node, TVID As String  
    If Len(HENArgs) > 0 Then  
        TVID = adhGetItem(HENArgs, "TVID")  
        For Each n In ExplorerPane.Nodes  
            If n.Key Like "*" & TVID & "*" Then  
                'found!  
                n.Selected = True  
                n.BackColor = vbHighlight  
                n.ForeColor = vbHighlightText  
                Call ExplorerPane_NodeClick(n)  
                Exit For  
            End If  
        Next  
    End If  
End Sub
```

Lidt forklaring til koden:

En global variabel HENArgs er i f.eks. søgeformularen tilført et argument (f.eks. "TVID" og en værdi (f.eks. "K31"). Husk at alle vores nodekeys er dannet af et bogstav og ID fra tabellen.. F.eks. er Key K31 er ID 31 fra tabellen TblKoncern.

Variablen TVID trækkes ud af den globale HENArgs og anvendes i søgningen gennem træet. Når og vis nøglen findes, sættes 'selected color' og klikfunktionen kaldes.

Vi når ikke her videre ind på, hvordan man opdaterer både eksplorerpane og de enkelte datapanes. Det er nemt nok at rette, slette og tilføje i højrepane, men hvordan får du opdateret træet og hvordan får du



tilføje en ny node af en bestemt type samtidig med at du får en frisk eksplorerPane.

Man sletter et knudepunkt(opdaterer træet) samtidig med at man sletter recorden ved at kalde følgende kode:

```
Private Sub BtnFjernnode_Click()  
    ExplorerPane.Nodes.Remove ExplorerPane.SelectedItem.Index  
End Sub
```

Hvordan springes til en ny node - det gør man f.eks. sådan:

På højrepanen kalder man f.eks. denne kode:

```
Private Sub BtnAddPost_Click()  
    nodekey = "F" & Me.FID ' F for firmanode over adressenoden  
    Call Form_RiskExplorer.Springtil_Ny_Node(nodekey)  
End Sub
```

I Hovedformularens code behind kaldes så:

```
Public Sub Springtil_Ny_Node(TVID)  
    Dim n As Node  
    If Len(TVID) > 0 Then  
        ExplorerPane.Nodes.Clear  
        LoadTree  
        For Each n In ExplorerPane.Nodes  
            If n.Key Like TVID Then  
                'found!  
                n.Selected = True  
                n.BackColor = vbHighlight  
                n.ForeColor = vbHighlightText  
                Call ExplorerPane_NodeClick(n)  
                Exit For  
            End If  
        Next  
    End If  
End Sub
```

Skal en ny node gemmes - kan man i gemmefunktionen anvende følgende kode for f.eks. adresser:

```
If Me.FId = 0 Then ' har vi ikke en moder Firma ID  
'hvis kun et niveau op - vi låner værdien af modernodens key  
    Set n = Me.Parent.ExplorerPane.SelectedItem.Parent  
' husk vi er et niveau længere nede hvis vi har indskudt en info node mellem firma/division og adresse og  
skal så bruge et ekstra Parent !  
    'Set n = Me.Parent.ExplorerPane.SelectedItem.Parent.Parent  
    Me.FId = Mid(n.Key, 2)  
End If
```

Ikke mere denne gang, der skal også være til en anden artikel -måske. Demodatabase vil blive uploadet et sted ved først givne lejlighed.

Henrik

#### **Kommentar af -anders- d. 25. Jan 2006 | 1**

Spændende artikel, der grundigt beskriver endnu flere muligheder med TreeView kontrollen. Tak for en flot demo.

#### **Kommentar af nih d. 19. Jan 2006 | 2**

Super artikel - glad for at du følger op på emnet :)

#### **Kommentar af rashid d. 18. Jan 2006 | 3**

god, men kunne godt tænke mig at se eksempeldatabasen! :-)

#### **Kommentar af huset d. 18. Jan 2006 | 4**

spændende men ufatteligt lang!

#### **Kommentar af brianboenk d. 28. Jan 2009 | 5**

Super cool artikel, selv jeg kunne følge med og lave min egen version til et mindre projekt... Tak for både demo og guide.