



TreeView i Access I

Dette er en artikel om, hvordan man kan bruge et TreeView til at præsentere data fra en Access database med flere en til mange relationer. Mit eksempel er en database over telefonnumre i virksomheder med flere afdelinger

Skrevet den **06. Feb 2009** af **nih** | kategorien **Databaser / Access** | ★★★★★

Rettelser:

10/08 2004 - Original udgave

16/08 2004 - Uddybende forklaringer

18/08 2004 - Små rettelser med hjælp fra Mugs :)

Baggrund

Et TreeView kan præsentere data fra flere en til mange relationer på en overskuelig måde. Mange gange ses formulare med 3 eller flere indlejrede underformulare, hvilket kan virke uoverskueligt.

Løsningen er et TreeView, som er opbygget ligesom Windows Stifinder. Brugeren kan folde træet ind og ud ved at klikke på de enkelte elementer kaldet noder. TreeViewet, der findes i Access er desværre ikke dataafhængigt - vi skal selv sørge for at træet er opdateret med de rigtige data fra vores tabeller/forespørgsler.

Mit eksempel er lidt tænkt, men viser principperne for hvordan træet loades med de korrekte data. Mit eksempel skitserer et eller flere firmaer med hver flere afdelinger der igen har flere ansatte med en masse telefoner.

Vi starter med tabellerne:

Tabelstruktur

Følgende tabeller oprettes

```
tFirma:   firmald, FirmaNavn
tAfdeling: afdld, firmald, afdNavn
tMedarbejdere: medArbld, afdld, medArbNavn
tTelfon:   tlfld, medArbld, tlfNr, art
```

Felterne siger sig selv på nær art, der kan anvendes til at angive om det er privat, mobil e.a.

Det er en fordel at relationerne sættes rigtigt, ellers vil koden fejle når f.eks en tAnsatt ikke er relateret til en tAfdeling.

Det vil være en god ide at indsætte nogle test data, så vi kan se resultatet af vores anstrengelser senere.

Formularen

Der oprettes en formular med:

-liste (Den almindelige fra værktøjskassen)

-et treeview værktøjslinje: værktøjskasse - flere kontrolelementer - Microsoft treeView Control

De navngives i egenskabsarket med hhv. lstTlf og ctlTree

I lstTlf sættes 'AntalKolloner' = 4 og

'KolloneBredde = 0cm;3cm;2cm;3cm

Koden

Så kommer det sjove ;o)

Vi starter med at loade data ind i træet når formularen åbnes.

Først slettes noderne i træet - dvs. de data der måtte være slettes

Så åbner vi den første tabel som et recordset og løber alle posterne igennem

For hver post oprettes en **rodNode** med navnet på firmaet som text og firmald som key. Firmaid'et sættes sammen med ordet "firma" så vi senere ved at noder med ordet "firma" i deres key kommer fra tabellen firma.

Derefter åbnes tabellen tAfdeling og på samme måde som gennemløbes tabellen og for hver post oprettes en **childNode**. Anvendes afdNavnet som nodens text og afdld som key OG den relateres til den Node i træet der har en key med værdien "firma" & firmald.

På formens VedIndlæsning lægges følgende kode ind:

```
Private Sub Form_Load()
```

```
    'Alle noder i træet fjernes
```

```
    ctlTree.Nodes.Clear
```

```
    Dim rs As New ADODB.Recordset
```

```
    ' åbner tabellen tFirma og tilføjer en rodNode for hver
```

```
    ' nodens key sættes = "firma" & firmald for senere at
```

```
    ' bruge den når der vælges en node i træet samt lave
```

```
    ' child noder
```

```
    rs.Open "tFirma", CurrentProject.Connection
```

```
    Do While Not rs.EOF
```

```
        ctlTree.Nodes.Add , , "firma" & rs!Firmald, rs!firmanavn
```

```
        rs.MoveNext
```

```
    Loop
```

```
    rs.Close
```

```
    ' vi fortsætter med at åbne medarbejdertabellen og
```

```
    ' tilføje en afdeling med reference til rodnodens
```

```
    ' key værdi = firma id
```

```

rs.Open "tAfdeling", CurrentProject.Connection
Do While Not rs.EOF
ctlTree.Nodes.Add "firma" & rs!Firmald, tvwChild, "afd" & rs!AfdID, rs!afdnavn
    rs.MoveNext
Loop
rs.Close

```

```

rs.Open "tMedarbejdere", CurrentProject.Connection
Do While Not rs.EOF
ctlTree.Nodes.Add "afd" & rs!AfdID, tvwChild, "medArb" & rs!medArbId, rs!medArbNavn
    rs.MoveNext
Loop
rs.Close
Set rs = Nothing

```

End Sub

Det var det nu tegnes træet rigtigt når formen åbnes :o)

Herefter skal listboxen fyldes med telefonnumre fra de firmaer, afdelinger eller ansatte brugeren klikker på.

treeWiets hændelse NodeClick kan anvendes til at finde ud af hvilken node vores bruger har klikket på (valgt).

Der oprettes en SQL sætning (evt cuttet fra forespørgsels designeren) som vi kan arbejde videre på.

Herefter tjekkes om nodens keyValue indeholder ordet firma, afd eller medarb.

Hvis f.eks key indeholder ordet "afd" så barberes de første 3 tegn væk of tilbage står afdId som sættes ind i SQL'en som en where del.

Så er det bare at opdatere listboxen med den nye SQL

'Hændelsen node_Click indeholder den node der klikkes på
'vi trækker værdien key ud og opdatere listboxen

```
Private Sub ctlTree_NodeClick(ByVal Node As Object)
```

```
    'dynamisk foresp til listboxen
```

```

    ssql = "SELECT tlfId, MedarbNavn, tlfArt, tlfNr " & _
"FROM (tAfdeling INNER JOIN tMedarbejdere ON tAfdeling.afdId = tMedarbejdere.afdId) INNER JOIN tTelfon ON tMedarbejdere.medarbId = tTelfon.medArbId "
```

```

'Hvis key indeholder: "firma" & nr
'ved jeg det er id fra tabellen tFirma.
""afd" & nr fra tAfdeling osv

```

```

If InStr(Node.Key, "firma") Then
ssql = ssql & "where firmald = " & Mid(Node.Key, 6)
Elseif InStr(Node.Key, "afd") Then
ssql = ssql & "where tAfdeling!afdId = " & Mid(Node.Key, 4)

```

```
ElseIf InStr(Node.Key, "medArb") Then
ssql = ssql & "where tMedarbejdere!medArbld = " & Mid(Node.Key, 7)
Else
    ssql = ""
End If
```

```
'Istboxen opdateres:
Me.IstTlf.RowSource = ssql
Me.IstTlf.Requery
```

End Sub

Jeg håber koden er kommenteret godt nok og i kan bruge dette lille eksempel til noget.

Jeg vil i næste udgave vise hvorledes man med en rekursiv procedure kan bygge et træ vha. en tabel tNode:Nodeld, ParrentId, Navn

Niels

Konstruktive kommentare er velkomne
- så kan jeg forhåbentlig forbedre min skrive færdigheder.

Kommentar af -anders- d. 19. Aug 2004 | 1

Har selv længe haft lyst til at benytte denne control som "navigator", glimrende eks. og super godt forklaret, har nu fået mod på at prøve denne control. Keep up the work.
Anders.

Kommentar af thomasjepesen d. 16. Aug 2004 | 2

Godt eksempel, som jeg glæder mig til at afprøve. Jeg har aldrig fået taget tiden til at afprøve Treeview-objektet, selvom jeg sagtens kan se mange muligheder i det.
Dog kan jeg ikke lade være at bemærke, at du bruger CurrentProject.Connection flere gange i stedet for at erklære et connection-objekt, ligesom indrykninger vil fremme læsevenligheden.

Kommentar af mugs d. 17. Aug 2004 | 3

Har længe haft kig på denne control, men ikke haft lejlighed til at sætte mig ind i den. Denne konstruktive og velformulerede artikel giver mig et velrettet spark i den rigtige retning. Vil anvende controllen til at styre medarbejderes placering i et hieraki med få afdelinger og mange underafdelinger

Kommentar af carstensurland d. 13. Aug 2004 | 4

Kommentar af markou d. 18. Oct 2004 | 5

Ligesom nedenstående kommentar, har jeg længe selv haft ønske om at benytte TreeView, men har aldrig fået det til at virke - det kan jeg nu ! Kanon artikel !

Kommentar af charlotterj d. 22. Nov 2004 | 6

God artikel