



ListView i Access del 2

Et ListView kan vise data fra tabeller og forespørgsler på en overskuelig og fleksibel måde. Ofte støder man ind i nogle begrænsninger hvis man benytter standardlisten, sortering, kolonnebreder e.t.c.

Skrevet den **06. Feb 2009** af **-anders-** I kategorien **Databaser / Access** | ★★★★★

Indledning:

Som gennemgået i Del 1 i denne føljeton om ListView-kontrollen kan vi nu få vist data i vores ListView, men strukturen er noget låst, man kan feks. ikke redigere data eller søge efter data med den indbygget søgefunktion i Access, dette er selvfølgelig ikke tilfredsstillende.

I denne artikel kikker vi på 3 ting:

1. Åbne en bagvedliggende form ved dobbeltklik på en post på ListVewet, så data kan redigeres.
2. Brug af ImageList-kontrollen, så det muligt at tilføje et valgfrit ikon til venstre for teksten i første kolonne.
3. Sempel fritekstsøgning i ListVewet.

Til forskel fra den første del af artiklen vil jeg i denne artikel tage udgangspunkt i min demodatabase, dette betyder at alle objektnavne samt feltnavne henviser til demobasen.

Hvis du efter at have læst artiklen ønsker at få tilsendt demobasen er du velkommen til at sende en mail til anj@bryggemail.dk, så vil jeg sende den hurtigst muligt.

Som det første skal du oprette en tabel med nedenstående felter samt datavalidering, navngiv tabellen tbltitler

TitelID: (Autonummering, sættes til primær nøgle)

Titel: (Tekstfelt, feltet skal være oblikatorisk, og tillad nullængde skal være sat til nej)

Kunster: (Tekstfelt, feltet skal være oblikatorisk, og tillad nullængde skal være sat til nej)

Genre: (Tekstfelt, feltet skal være oblikatorisk, og tillad nullængde skal være sat til nej)

Nu til afsnittet hvor vi skal til at rediger i data:

Da ListVewet ikke fungerer på samme måde som en standard fortløbende formular, er det nødvendigt at have en form der viser indholdet af den post vi dobbeltklikker på i ListVewet.

Først skal du oprette en formular som enkeltkolonne, i designvisning sætter du postkilden til tbltitler, og tilføjer derefter alle felterne på formen, navngiv formen frmRedigerData

Opret igen en ny formular, denne form skal ikke have nogen postkilde, da den skal bruges til at vise listviewet, navngiv formularen frmListViewDel2.

I designvisning tilføjer du nu et listview som beskrevet i Del 1, du skal tilføje 3 kolonner med passende bredde, navngiv kolonnerne Titel,Kunster,Genre, navngiv listviewet til ListView2

Opret derefter et modul og kopier nedenstående kode ind i modulet, navngiv modulet molistdata (du kan navngive modulet til noget andet, i denne sammenhæng er det ligegyldigt hvad modulet hedder).

'Kode der skal kopieres ind i modulet:'

```
Function GetDataListView2()
```

```
Dim cn As ADODB.Connection
```

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

Dim itmX As ListItem

```
Set rs = New ADODB.Recordset
With rs
Set .ActiveConnection = CurrentProject.Connection
'Poster hentes fra den angivne tabel
.Open "Select * From tbltitler", , , adLockOptimistic
'Recordset løbes igennem til sidste post'
Do Until .EOF

'Data tilføjes i første kolonne, altså kolonne 0, denne kolonne indeholde titlen'
Set itmX = Form_frmListViewDel2.ListView2.ListItems.Add(, , rs!Titel)
'Herefter tilføjes data i de andre kolonner fra den angivne tabel'
itmX.SubItems(1) = rs!Kunster 'feltet kunster fra tbltitler
itmX.SubItems(2) = rs!Genre 'feltet genre fra tbltitler
.MoveNext
Loop
End With
'Recordset lukkes'
rs.Close
'Code end'
End Function
```

Åbn derefter formen frmListViewDel2 i designvisning, under hændelsen ved åbning skriver du Call GetDataListView2 ,når formen åbnes i normal visning kan du nu se de data du har indtastet i tbltitler.

Visning af post man har dobbeltklikket på i ListViewet:

For at åbne frmRedigerData og vise den aktuelle post er det nødvendigt at kende værdien af teksten i første kolonne altså Titlen på sangen, derfor har vi brug for en variable der kan aflæse værdien og gemme den, gør følgende for at oprette denne variable:

Åbn formen frmListViewDel2 i designvisning, gå derefter i VBA-editoren, helt øverst lige under Option Explicit skriver du Private sltem As String, i bunden af VBA-editoren kopiere du følgende ind:

```
Private Sub ListView2_ItemClick(ByVal Item As Object)
sltem = Item.Text
'Code end'
End Sub
```

Det var det, nu kender variabelen sltem værdien af første kolonne når du klikker på en post, kopier nu følgende ind i VBA-editoren:

```
Private Sub ListView2_DblClick()
DoCmd.OpenForm "frmRedigerData", , , "Titel = '" & sltem & "'"
'Code end'
End Sub
```

Sådan når du nu dobbeltklikker på en post på ListViewet åbnes frmRedigerData og viser den post du har dobbeltklikket på. Hvis der er to titler der er helt ens skrevet er det selvfølgelig et problem, alternativt kan man i stedet bruge TitleID som første kolonne, dette kræver så at der skal oprettes en kolonne mere i ListViewet. og koden til at hente data skal tilrettes.

Brug af ImageList til at håndtere ikon til venstre for teksten i første kolonne:

Her vil jeg beskrive hvordan du kan pifte dit design lidt op på ListViewet, om man gider bruger denne funktion eller ej er helt op til en selv det er udelukkende et spørgsmål om design/udsende, det har ingen

betydning for dataindholdet. Personligt syntes jeg at det meget sjovt at lege med o:).

Som sagt for at kunne håndtere dette har vi brug for en ImageList-kontrol, denne kontrol er også en ActiveX-kontrol som kalder samme OCX-fil som ListViewet, det er derfor ikke nødvendigt at lave en ny ref. i VBA-editoren. Gør følgende for at oprette en ImageList-kontrol på din ListView form:

Åbn formen frmListViewDel2 i designvisning, klik på værktøjskassen oppe i værktøjslinjen, klik derefter på værktøjsikonet nederst. Scroll ned til du finder Microsoft Imagelist Control version 6.0, (Der kan være andre versioner ,men som udgangspunkt skal du vælge samme Versionsnummer som dit ListView), træk nu kontrollen over på din form, det ligemeget hvor du placere kontrollen da den ikke kan ses i alm. visning, navngiv kontrollen ImageList2. Luk formen og svar ja til at gemme.

Åbn nu frmListViewDel2 i designvisning og højreklik på ImageList-kontrollen, og vælg ImageListCtrl-Objekt, klik derefter på Properties, nu vises egenskabsarket for Imagelist-kontrollen.

På fanebladet Generelt skal du sætte flueben i UseMaskColor, vælg derefter 16x16, det vigtigt at disse egenskaber sættes før vi tilføjer ikoner til ImageListen, det rimligt bøvet at ændre dem efter der er knyttet ikoner til Imagelisten. Klik nu på fanebladet Images, det er her vi tilføjer de ikoner vi vil arbejde med.

Klik på knappen Insert Picture, gennemse din PC efter Ikoner (.ico filer), og indsæt dem du ønsker at bruge, der skal mindst være valgt et før det virker. Du kan have ligeså mange ikoner på Imagelisten du ønsker. Hvert ikon du tilføjer for et Indexnummer, det er dette nummer vi skal bruge senere på ListViewet, klik på Ok for at lukke egenskabsarket. Højreklik derefter på ListViewet og aktiver egenskabsarket, klik på fanebladet Image List, i rulleboksen Normal og Small vælger du din Imagelist. Luk formularen og svar ja til at gemme.

Det var det nu er ListView og ImageList-kontrollerne "knyttet sammen" og der er tilføjet et eller flere ikoner til Imagelist-kontrollen, så nu mangler der bare koden til at loade det valgte ikon (indexnummer) til ListViewet når frmListViewDel2 åbnes, koden til at loade data er næsten magen til den første kode, det er kun en linje der er anderledes i stedet for:

```
Set itmX = Form_frmListViewDel2.ListView2.ListItems.Add( , rs!Titel)
```

Skal der stå:

```
Set itmX = Form_frmListViewDel2.ListView2.ListItems.Add( , rs!Titel,1,1)
```

Her vises altså det ikon der har indexnummer 1, hvis man ønsker at bruge et andet ikon fra Imagelisten ændre man blot indexnummer i koden.

Hvis man ikke ønsker at benytte sin Imagelist skal du blot fjerne indexnummer i koden, så vises ikonet ikke. Det kan måske virke lidt omstændigt at benytte ImageList-kontrollen, men i virkeligheden er det ikke så slemt at have med at gøre når først man har prøvet det et par gange.

Bemærk at de ikoner du indsætter i Imagelisten er intigreret, dvs. at du ikke behøver at angive nogen filsti til ikonerne efter de er indsat i kontrollen. Du skal ikke være nervøs for at størelsen på databasen vokser uhæmmet, dette sker ikke, kontrollen håndtere ikke disse ikoner på samme måde som et OLE-objekt felt i en tabel.

Fritekstsøgning i ListView-kontrollen:

Til slut vil jeg beskrive hvordan man kan fortage simpel fritekstsøgning i ListView-kontrollen, den indbygget søgefunktion (kikkerten) virker ikke her så vi er nødt til at lave den selv.

Du skal nu oprette en ny formular som er magen til frmListViewDel2, navngiv formularen frmListViewDel2Fritekstsogning, du skal gøre plads til en ubunden tekstboks og en kommandoknap øverst

i formen. Du kan med fordel tage en kopi af frmListViewDel2, det meste af koden kan genbruges blot du ændre objekt og feltnavne til det korrekte. Når formularen er oprettet skal du navngive ListViewet til ListView3. Hvis du har kopieret formularen skal al koden under hændelsen Ved åbning fjernes, da data først skal vises når søgning aktiveres, og ikke ved åbning af formen.

Åbn frmListViewDel2Fritekstsogning i designvisning og indsæt en ubunden tekstboks øverst i formen navngiv tekstboksen til txtFind, indsæt derefter en kommandoknap ved siden af tekstboksen navngiv knappen cmdfind

Det var formen med de tilhørende objekter, nu skal koden til frittekstsøgningen skrives, den er en smule anderledes end den første kode, da det er nødvendigt at tømme ListViewet for data hvergang der søges, men forskellen er ikke stor. Åbn dit modul og indsæt nedenstående kode:

'Bemærk at der ikke benyttes * som wilcard, da dette ikke kan bruges i ADO'

```
Function GetDataSogning()
```

'Her tømmes listviewet for data før nye data loades, dette er nødvendigt for ikke at loades de samme data igen og igen'

```
Form_frmListViewDel2Fritekstsogning.ListView3.ListItems.Clear
```

```
Dim cn As ADODB.Connection
```

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

```
Dim itmX As ListItem
```

```
'Variable til SQL strengen'
```

```
Dim sSQL1 As String
```

```
Set rs = New ADODB.Recordset
```

```
With rs
```

```
Set .ActiveConnection = CurrentProject.Connection
```

```
'Søgning igennem alle foruddefineret felter'
```

```
sSQL1 = "Select * From tbltitler where [Titel]&[Kunster]&[Genre] LIKE '%" &
```

```
Form_frmListViewDel2Fritekstsogning!txtFind & "%'"
```

```
'Poster hentes fra den angivne tabel
```

```
.Open sSQL1, , , adLockOptimistic
```

```
Do Until .EOF
```

```
'Her loades så de poster til listviewet som søgningen returnerer'
```

```
Set itmX = Form_frmListViewDel2Fritekstsogning.ListView3.ListItems.Add(, , rs!Titel)
```

```
'Herefter tilføjes data i de andre kolonner fra den angivne tabel der matcher søgningen'
```

```
itmX.SubItems(1) = rs!Kunster
```

```
itmX.SubItems(2) = rs!Genre
```

```
.MoveNext
```

```
Loop
```

```
End With
```

```
'Recordset lukkes'
```

```
rs.Close
```

```
'Code end'
```

```
End Function
```

Gem dine ændringer og luk modulet.

Åbn frmListViewDel2Fritekstsogning i designvisning under hændelsen Ved klik på cmdfind skriver du Call GetDataSogning

Når du åbner frmListViewDel2Fritekstsogning vises der ingen poster i ListViewet, du har jo fjernet koden ved hændelsen Ved åbning. Data vil først vises når du klikker på cmdfind, hvis du klikker på knappen uden at der er indtastet noget i tekstboksen txtFind vises alle poster, men indtaster du noget søgetekst, og derefter klikker på knappen vises kun de poster der matcher hele eller dele af søgeteksten.

Denne type frittekstsøgning er måske nok lidt mangelfuld, den er ikke speciel dynamisk, hvis man ændre sine kolonner i listviewet feks. ved at tilføje flere kolonner skal man jo direkte ind i koden og definere de ekstra felter, men det er hvad mine evner pt. strækker sig til, men et stykke af vejen er den, syntes jeg anvenlig nok.

Koden til at frittekstsøge drillede mig en del i begyndelsen da jeg brugte * som wildcard og ikke %, efter en ihærdig indsats fra terry her på eksperten fandt jeg dog ud af det, så tak til terry.

Afslutning:

Som nævnt i indledningen i del 1 er artiklen ikke en anbefaling til at bruge det ene frem det andet, men udelukkende tænkt som alternativ/inspiration. ListViewet har fordele, men bestemt også bagdele når man arbejder med det i Access-miljøet, grundlæggende er kontrollen en VB-kontrol, og bliver benyttet meget af udviklere i VB-miljøet, men kan også benyttes i Access-miljøet et stykke af vejen.

Hvis du kikker i VB-kategorien her på eksperten og skriver søgeordet ListView vil du få en del resultater, der er mange gode indlæg omkring kontrollen, og man kan sagtens lære af disse indlæg, så jeg vil varmt anbefale at prøvet det hvis du vil vide mere om ListView-kontrollen.

Det var hvad jeg havde valgt bringe om ListViewet-kontrollen i denne artikel, jeg håber du for lyst til at prøve denne kontrol da den mægtig sjov at arbejde med. God fornøjelse.

Mvh. Anders.

Kommentar af terry d. 24. May 2005 | 1

Once again a good article Anders. Hope it encourages others to use both the TreeView and ListView controls in Access applications.

Kommentar af mikkellk d. 01. Apr 2005 | 2

Endnu en glimrende artikel om listview kontrollen.

Kommentar af mafia666 d. 03. Apr 2005 | 3