

## 8.2. Vidd, dobd, vagy illesztd be!

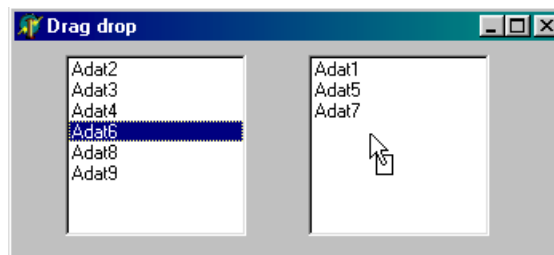
1. Listák közötti adatcsere a [DragDrop](#) használatával
2. Igazodó vezérlősor [DragDock](#)



Készítsünk egy olyan alkalmazást, amely két listaablakot jelenít meg és mindkét ablakból a listaelemek "fogd és vidd" megoldással áthelyezhetők a másikba. (*DragDrop*)

Tervezéskor a formra két listaablakot helyezünk, melyek "fogd és vidd" jellemzőit a form létrehozásakor állítjuk be:

```
procedure TForm1.FormCreate(Sender: TObject);
var i:integer;
begin
  // mindkét lista automatikus "fogd és vidd" tulajdonságú
  listbox1.dragmode:=dmAutomatic;
  listbox2.dragmode:=dmAutomatic;
  // a bal oldali listaablak feltöltése
  For i:=1 to 9 do
    listbox1.items.add('Adat'+inttostr(i));
  end;
```



Mindkét listaablak esetén ugyanazzal az eseménykezelővel dolgozzuk fel az áthúzást, illetve a ledobást, hiszen a *Sender* paraméter megadja azt a komponenst amelyik az üzenetet küldi, és a *Source* pedig azt, amelyiket mozgatjuk.

```
procedure TForm1.ListBoxDragOver(Sender, Source: TObject;
                                X, Y: Integer; State: TDragState;
                                var Accept: Boolean);
begin
  // mindkét listaablakban az áthúzás kezelése
  if (TListBox(source).Itemindex<0) then
    TListBox(source).dragcursor:=crNoDrop;
end;

procedure TForm1.ListBoxDragDrop(Sender, Source: TObject;
                                X, Y: Integer);
begin
  // mindkét listaablakban a ledobás kezelése
  if (TListBox(source).Itemindex>-1) then
  begin
    TListBox(sender).Items.Add(TListBox(source).items
                               [TListBox(source).Itemindex]);
    TListBox(source).items.Delete(TListBox(source).Itemindex);
  end;
end;
```

Ha a listában nincs kiválasztott listaelem (például üres lista), akkor tiltjuk a "fogd és vidd" működést:

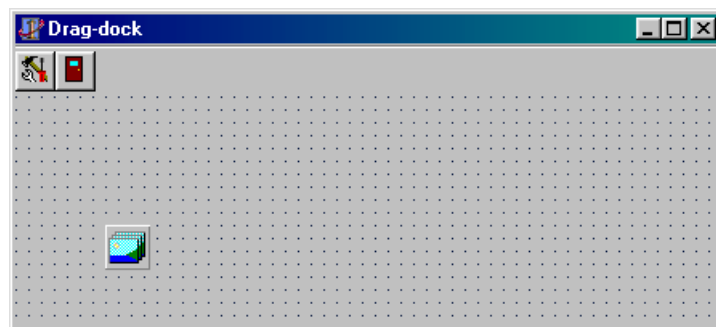
```
procedure TForm1.ListBoxClick(Sender: TObject);
begin
  if TListBox(Sender).Itemindex>-1 then
    TListBox(Sender).dragcursor:=crDrag;
end;
```



Készítsünk olyan alkalmazást, melynek vezérlőszora az ablak mind a négy oldalához igazítható, mint a képernyőn a tálca! Legyen a vezérlősor leszakítható az ablakról! (*DragDock*)

Készítsük elő a formot! A *ToolBar1* lesz a mozgatható vezérlősor. Ennek megfelelően **DragMode** tulajdonsága *dmAutomatic* és **DragKind** jellemzője pedig *dkDock*. Helyezzünk el két gombot az eszközsoron! Az első (*ToolButton1*) arra szolgál, hogy beállítsuk, hogy az eszközsor leszakítható-e. Ezért ennek **Style** tulajdonsága *tpsCheck*. A másik gombbal kiléphetünk az alkalmazásból. A képeket az *ImageList1*-be töltjük és ezt a *ToolBar1*-hez kötjük annak **Images** tulajdonsága segítségével. Ezek után a gombok **ImageIndex** tulajdonsága a képlistából jelöli ki a megfelelő képet.

Felteszünk a formra, a *Jobb*, a *Bal*, a *Lent* és a *Fent* paneleket a megfelelő **Align** tulajdonságok *alRight*, *alLeft*, *alBottom* és *alTop*. Az első két panel szélességét, az utóbbi kettő magasságát 0-ra állítjuk. Mindegyik **AutoSize** tulajdonsága *true*. Ezek lesznek azok a konténerek, melyekhez az eszközsort csatlakoztatjuk, ezért mind a négy **DockSite** tulajdonsága *true*, míg ugyanez a form esetén *false*.

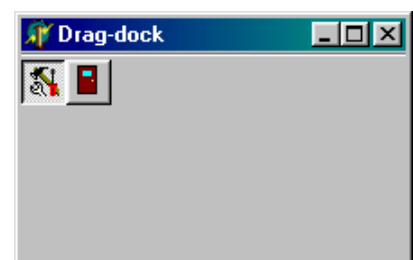


A program indításakor nem engedjük leszakítani az eszközsort, felengedjük a leszakíthatóságot engedélyező gombot és csatlakoztatjuk az eszközsort a *Fent* elemhez.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  ToolBar1.FloatingDockSiteClass:=nil;
  ToolButton1.Down:=false;
  ToolBar1.ManualDock(Fent, nil, alNone);
end;
```

Ha a *ToolButton1* gomb lenyomott, akkor önálló ablak segítségével az eszközsor leszakítható, egyébként csak a széleken lehet csatlakoztatni.

```
procedure TForm1.ToolButton1Click(Sender: TObject);
begin
  // Leszakítható, vagy sem
  if ToolButton1.Down then
    ToolBar1.FloatingDockSiteClass:=TCustomDockForm
  else
    ToolBar1.FloatingDockSiteClass:=nil;
end;
```



Az alkalmazásból való kilépés a másik eszközgomb feladata.

```
procedure TForm1.ToolButton2Click(Sender: TObject);
begin
  Application.Terminate;
end;
```

A négy panel esetén ugyanazt az eljárást adjuk meg az **OnDockOver** esemény kezelésére. Az eseménykezelő eljárásban mind a négy elemre kiszámoljuk a megfelelő oldalhoz igazodó eszközsor megjelenítéséhez, szükséges méretű téglalapot és ezt átadjuk a **DockRect** tulajdonságnak.

```

procedure TForm1.LentDockOver(Sender: TObject; Source: TDragDockObject;
  X, Y: Integer; State: TDragState; var Accept: Boolean);
var a:TRect;
      p:TPoint;
begin
  p.x:=TControl(Sender).left;
  p.y:=TControl(Sender).top;
  p:=Form1.ClientToScreen(p);
  // Ha lent vagy fenn van a csatlakozás
  if (sender=lent) or (sender=fent) then
    begin
      a.left:=p.x;
      if Sender=lent then
        a.top:=p.y-magassag
      else
        a.top:=p.y;
        a.bottom:=a.top+Magassag;
        a.right:=a.left+TControl(Sender).width;
      end
    end
  else
    begin
      // Ha jobb vagy baloldalon van a csatlakozás
      if (sender=bal) then
        a.left:=p.x
      else
        a.left:=p.x-magassag;
        a.top:=p.y;
        a.bottom:=a.top+TControl(Sender).height;
        a.right:=a.left+Magassag;
      end;
      Source.DockRect:=a;
    end;
end;

```

