How to create a TListBox with Drag and Drop capabilities

Question

How do I drag and drop an item from one position in a listbox to another manually? That is, the user can determine the order of the items.

```
unit PBReorderListBox;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TPBReorderListBox = Class(TListBox)
  private
    FDragIndex: Integer;
    FDragImage: TDragImagelist;
  protected
    procedure DoStartDrag(var DragObject: TDragObject); override;
    procedure DragOver(Source: TObject; X, Y: Integer;
      State: TDragState; var Accept: Boolean); override;
  public
    procedure DefaultDragOver(Source: TObject; X, Y: Integer;
      State: TDragState; var Accept: Boolean); virtual;
    procedure DefaultStartDrag(var DragObject: TDragObject); virtual;
    procedure DefaultDragDrop(Source: TObject; X, Y: Integer); virtual;
    procedure CreateDragImage(const S: String);
    procedure DragDrop(Source: TObject; X, Y: Integer); override;
    function GetDragImages: TDragImagelist; override;
    property DragIndex: Integer read FDragIndex;
    property DragImages: TDragImageList read GetDragImages;
end;

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('PBGoodies', [TPBReorderListBox]);
end;

procedure TPBReorderListBox.CreateDragImage(const S: String);
var
  size: TSize;
  bmp: TBitmap;
begin
  if not Assigned(FDragImage) then
    FDragImage := TDragImagelist.Create(self)
  else
    FDragImage.Clear;
  Canvas.Font := Font;
  size := Canvas.TextExtent(S);
  FDragImage.Width := size.cx;
  FDragImage.Height := size.cy;
  bmp := TBitmap.Create;
  try
    bmp.Width := size.cx;
    bmp.Height := size.cy;
    bmp.Canvas.Font := Font;
    bmp.Canvas.Font.Color := clBlack;
    bmp.Canvas.Brush.Color := clWhite;
    bmp.Canvas.Brush.Style := bsSolid;
    bmp.Canvas.TextOut(0, 0, S);
    FDragImage.AddMasked(bmp, clWhite);
  finally
    bmp.free
  end;
  ControlStyle := ControlStyle + [csDisplayDragImage];
end;

procedure TPBReorderListBox.DefaultDragDrop(Source: TObject;
  X, Y: Integer);
var
  dropindex, ti: Integer;
  S: String;
  obj: TObject;
begin
  if Source = Self then
  begin
    S := Items[FDragIndex];
    obj := Items.Objects[FDragIndex];
    dropIndex := ItemAtPos(Point(X, Y), True);
    ti := TopIndex;
    if dropIndex > FDragIndex then
      Dec(dropIndex);
    Items.Delete(FDragIndex);
    if dropIndex < 0 then
      items.AddObject(S, obj)
    else
      items.InsertObject(dropIndex, S, obj);
    TopIndex := ti;
  end;
end;

Procedure TPBReorderListBox.DefaultDragOver(Source: TObject;
  X, Y: Integer; State: TDragState; Var Accept: Boolean);
begin
  Accept := Source = Self;
  if Accept then
  begin
    {Handle autoscroll in the "hot zone" 5 pixels from top or bottom of
    client area}
    if (Y < 5) or ((ClientHeight - Y) <= 5) then
    begin
      FDragImage.HideDragImage;
      try
        if Y < 5 then
        begin
          Perform(WM_VSCROLL, SB_LINEUP, 0);
          Perform(WM_VSCROLL, SB_ENDSCROLL, 0);
        end
        else
        if (ClientHeight - Y) <= 5 then
        begin
          Perform(WM_VSCROLL, SB_LINEDOWN, 0);
          Perform(WM_VSCROLL, SB_ENDSCROLL, 0);
        end;
      finally
        FDragImage.ShowDragImage;
      end;
    end;
  end;
end;

procedure TPBReorderListBox.DefaultStartDrag(var DragObject: TDragObject);
begin
  FDragIndex := ItemIndex;
  if FDragIndex >= 0 then
    CreateDragImage(Items[FDragIndex])
  else
    CancelDrag;
end;

procedure TPBReorderListBox.DoStartDrag(var DragObject: TDragObject);
begin
  if Assigned(OnStartDrag) then
    inherited
  else
    DefaultStartDrag(DragObject);
end;

procedure TPBReorderListBox.DragDrop(Source: TObject; X, Y: Integer);
begin
  if Assigned(OnDragDrop) then
    inherited
  else
    DefaultDragDrop(Source, X, Y);
end;

procedure TPBReorderListBox.DragOver(Source: TObject; X, Y: Integer;
  State: TDragState; var Accept: Boolean);
begin
  if Assigned(OnDragOver) then
    inherited
  else
    DefaultDragOver(Source, X, Y, State, Accept);
end;

function TPBReorderListBox.GetDragImages: TDragImagelist;
begin
  Result := FDragImage;
end;

end.
```

Original resource:	The Delphi Pool
Author:	Peter Below
Added:	2009-10-26
Last updated:	2009-10-26