

# How to create a TListBox that is resizable at runtime

## QUESTION

How can I let my user resize a TListBox (using the mouse) at runtime, like the way we do at design time in Delphi?

Here's a sample listbox control that can be resized at runtime. It has two new properties: *AllowResize* (True by default), and *ResizeBorder* (shows a resizing border if True). You can still resize the box if *ResizeBorder* is False (as long as *AllowResize* is True), but only the standard listbox border is displayed. Since the control can now be resized, the *Constraints* property is also published.

```
unit ResizeListBox;

interface

uses
  Windows, Messages, SysUtils, Classes, Controls, StdCtrls;

type
  TResizeListBox = class(TListBox)
  private
    FAllowResize: Boolean;
    FResizeBorder: Boolean;
    procedure SetAllowResize(const Value: Boolean);
    procedure SetResizeBorder(const Value: Boolean);
    procedure WMNCHitTest(var Msg: TWMNCHitTest); message WM_NCHITTEST;
  protected
    procedure CreateParams(var Params: TCreateParams); override;
  public
    constructor Create(AOwner: TComponent); override;
  published
    property AllowResize: Boolean
      read FAllowResize write SetAllowResize default True;
    property ResizeBorder: Boolean
      read FResizeBorder write SetResizeBorder default False;
    property Constraints;
  end;

implementation

{ TResizeListBox }

constructor TResizeListBox.Create(AOwner: TComponent);
begin
  inherited Create(AOwner);
  FAllowResize := True;
end;

procedure TResizeListBox.CreateParams(var Params: TCreateParams);
begin
  inherited CreateParams(Params);
  if FAllowResize then
    Params.WindowClass.style := Params.WindowClass.style
      or (CS_VREDRAW + CS_HREDRAW);
  if FResizeBorder then
    Params.Style := Params.Style or WS_THICKFRAME;
end;

procedure TResizeListBox.SetAllowResize(const Value: Boolean);
begin
  FAllowResize := Value;
  RecreateWnd;
end;

procedure TResizeListBox.SetResizeBorder(const Value: Boolean);
begin
```

```

    FResizeBorder := Value;
    RecreateWnd;
end;

procedure TResizeListBox.WMNCHitTest(var Msg: TWMNCHitTest);
var
    P: TPoint;
begin
    inherited;
    if FAllowResize then
    begin
        if (not FResizeBorder) and (Msg.Result = HTBORDER) then
        begin
            {Cursor is on the normal border - determine where}
            P := Parent.ScreenToClient(Point(Msg.XPos, Msg.YPos));
            if PtInRect(Rect(Left, Top, Left + 8, Top + 8), P) then
                Msg.Result := HTTOPLEFT
            else
                if PtInRect(
                    Rect(Left + Width - 8, Top, Left + Width, Top + 8), P
                ) then
                    Msg.Result := HTTOPRIGHT
                else if PtInRect(
                    Rect(Left, Top + Height - 8, Left + 8, Top + Height), P
                ) then
                    Msg.Result := HTBOTTOMLEFT
                else if PtInRect(
                    Rect(
                        Left + Width - 8, Top + Height - 8, Left + Width, Top + Height
                    ),
                    P
                ) then
                    Msg.Result := HTBOTTOMRIGHT
                else if PtInRect(Rect(Left, Top, Left + 8, Top + Height), P) then
                    Msg.Result := HTLEFT
                else if PtInRect(Rect(Left, Top, Left + Width, Top + 8), P) then
                    Msg.Result := HTTOP
                else if PtInRect(
                    Rect(Left + Width - 8, Top, Left + Width, Top + Height), P
                ) then
                    Msg.Result := HTRIGHT
                else
                    Msg.Result := HTBOTTOM;
            end;
        end
        else if Msg.Result in [
            HTBOTTOM, HTBOTTOMLEFT, HTBOTTOMRIGHT, HTLEFT,
            HTRIGHT, HTTOP, HTTOPLEFT, HTTOPRIGHT
        ] then
            Msg.Result := HTBORDER; {Prevent resize if no AllowResize is False}
        end;
    end.
end.

```

Original resource:	The Delphi Pool
Author:	Yorai Aminov
Added:	2009-10-26
Last updated:	2009-10-26