How to define a minimum size for a component

> ***Question***
>
> I have a component (derived from *TGraphicControl*) and the user should only be able to resize the component to a minimum size or higher. How do I have to implement the *SetHeight* and *SetWidth* procedures?

You can override the *SetBounds* method. Since this method is responsible for setting new coordinates and dimensions for the control. Here's an example:

```
TMyGraphicControl = class(TGraphicControl)
  public
    procedure SetBounds(ALeft: Integer; ATop: Integer; AWidth:
      Integer; AHeight: Integer); override;
  end;

const
  min_Widht = 20;
  min_Height = 20;

procedure TMyGraphicControl.SetBounds(ALeft: Integer; ATop: Integer;
  AWidth: Integer; AHeight: Integer);
begin
  if AWidth < min_Widht then
    AWidth := min_Widht;
  if AHeight < min_Height then
    AHeight := min_Height;
  inherited SetBounds(ALeft, ATop, AWidth, AHeight);
end;
```

You may also take a look at the *CanResize* method. It is called automatically when an attempt is made to resize the control, after any autosizing has occurred. You can override it and return false in the result, in case the new height or width is less than the minimum value.

```
type
  TMyGraphicControl = class(TGraphicControl)
  protected
    function CanResize(var NewWidth: Integer;
      var NewHeight: Integer): Boolean; override;
  end;

function TMyGraphicControl.CanResize(var NewWidth: Integer;
  var NewHeight: Integer): Boolean;
begin
  if not (csLoading in ComponentState) and ((NewWidth < min_Width)
  or (NewHeight < min_Height)) then
    Result := false
  else
    Result := inherited CanResize(NewWidth, NewHeight);
end;
```

Alternative Method

Montor suggests the following solution:

```
with MyControl.Constraints do
begin
  MaxHeight :=100;
  MaxWidth  :=100;
  MinHeight :=50;
  MinWidth  :=50;
end;
```

Original resource: The Delphi Pool
Author: Serge Gubenko

Added:	2009-08-12
Last updated:	2011-01-16