

# AbMask - Tutorial

This demo shows how to use the Abakus Mask controls. This are very easy examples but they will help to understand how they work and how to use.  
Best is, to try all examples... this will take only few minutes

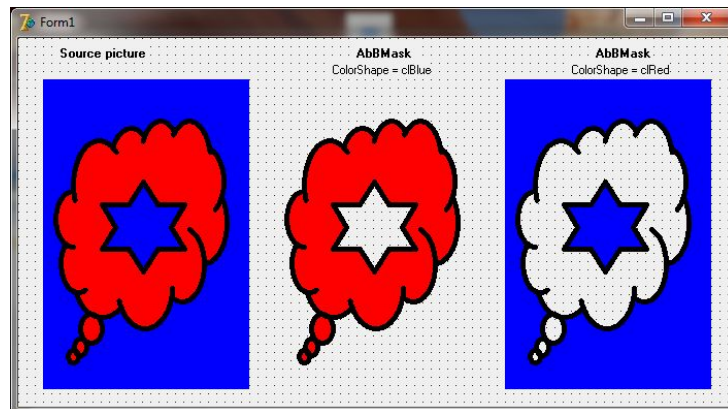
Actually all demo code is in Delphi but in such a easy form that it should not be a problem to convert it to C++Builder.

Info: All sample pictures used here you will find in the folder `..\Abakus\Demos\Images\`

## Basics

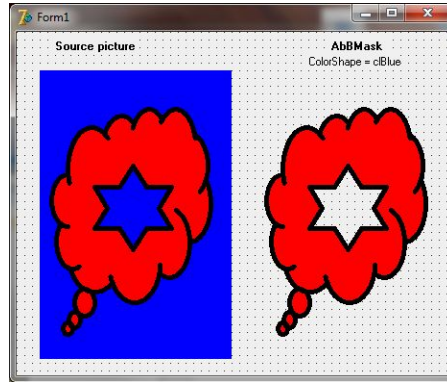
To “mask” the control all parts you want to exclude from the shape should be in the same color. By default `clBlue` is used to mask but any other color can be selected.

Info: The best result you will get if antialiasing is not used in this areas.





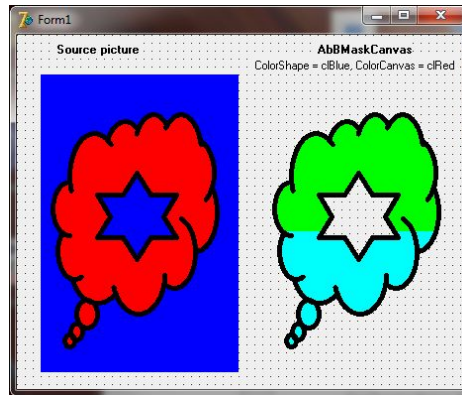
## TabBMask first try..



1. Create a new VCL application
2. select a TAbBMask control from the Toolbar->AbMisc and place this on the form
3. click to the Picture property and select the above image from the image folder  
(..\Abakus\Demos\Images\)
4. set the Cursor of the control to any other than default, this is to show you the shape  
when moving with the mouse over the control in runtime
5. run and move with the mouse over the control....
6. may change the ColorShape to clRed and repeat step 5



## TabBMaskCanvas first try..



1. Create a new VCL application
2. select a TAbBMaskCanvas control from the Toolbar->AbMisc and place this on the form
3. click to the Picture property and select the above image from the image folder (..\Abakus\Demos\Images\)
4. the red colored area of the image represents the “user canvas” (default clRed, any other color is selectable by property ColorCanvas)
5. by default the “user canvas” is filled in clLime and clAqua

You can paint to this “user canvas” by adding some lines of code to the OnChange event of the component. The size of the canvas results of the bounding rectangle around the ColorCanvas (default clRed).

All painting to this “user canvas” is double buffered to avoid flickering.

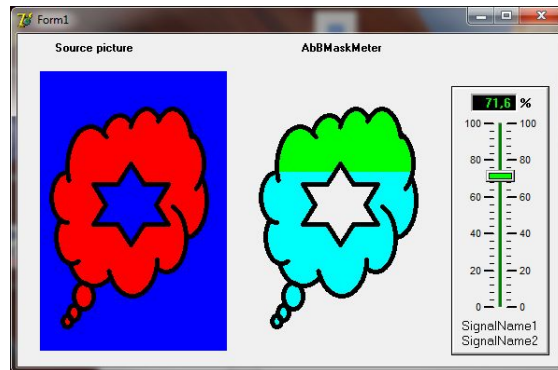
### 6. Code example

```
procedure TForm1.AbBMaskCanvas1Change(can: TCanvas);  
var  
    w, h : Integer;  
begin  
    // get the width and height  
    w := can.ClipRect.Right;  
    h := can.ClipRect.Bottom;  
  
    // fill the total area in clNavy  
    can.Brush.Style := bsSolid;  
    can.Brush.Color := clNavy;  
    can.Pen.Color   := clNavy;  
    can.Pen.Style   := psSolid;  
    can.Rectangle(Rect(0,0,w,h));  
end;
```

If the OnChange event is assigned then you are responsible to draw the “customer canvas”



## TAbsMaskMeter, first try..



1. Create a new VCL application
2. select a TAbsMaskMeter control from the Toolbar->AbAnalog and place this on the form
3. click to the Picture property and select the above image from the image folder (..\Abakus\Demos\Images\)
4. add a AbVSlider from the Toolbar->AbAnalogAdj to the form
5. Add to the AbVSlider OnValueChanged event this line of code to link the AbVSlider.Value with the AbBMaskMeter.Value  
`AbBMaskMeter1.Value := AbVSlider1.Value;`
6. Run the application and move the slider....

Like on AbBMaskCanvas if the OnChange is assigned then you are responsible to draw the "User canvas".

A single-type variable "PPH" is available, this variable shows the component.value in percent.

Now we will use the above example, the OnChange Event, and some lines of code to change the "Fill-direction" to horizontal...

Code example:

```
procedure TForm1.AbBMaskMeter1Change(can: TCanvas);
var
    w, h, cpos : Integer;
begin
    // get the width and height
    w := can.ClipRect.Right;
    h := can.ClipRect.Bottom;

    // fill the total area in clNavy
    can.Brush.Style := bsSolid;
    can.Brush.Color := AbBMaskMeter1.ColBackground;
    can.Pen.Color    := AbBMaskMeter1.ColBackground;;
    can.Pen.Style    := psSolid;
    can.Rectangle(Rect(0,0,w,h));

    // set the bar color
```

```

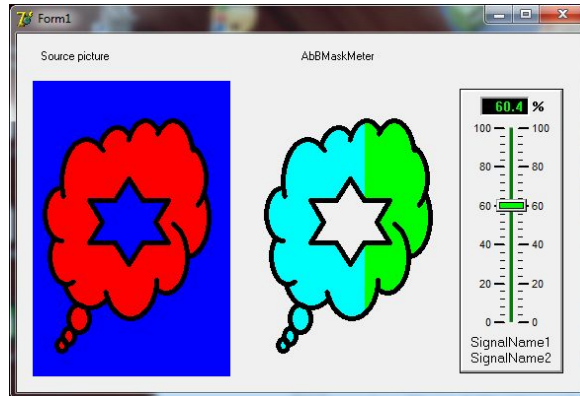
can.Brush.Color := AbBMaskMeter1.ColForeground;
can.Pen.Color   := AbBMaskMeter1.ColForeground;

// calculate the Position according PPH (Value in percent)
cpos := round((w/100) * AbBMaskMeter1.PPH);

// draw the bar rectangle
can.Rectangle(Rect(0,0,cpos,h));
end;

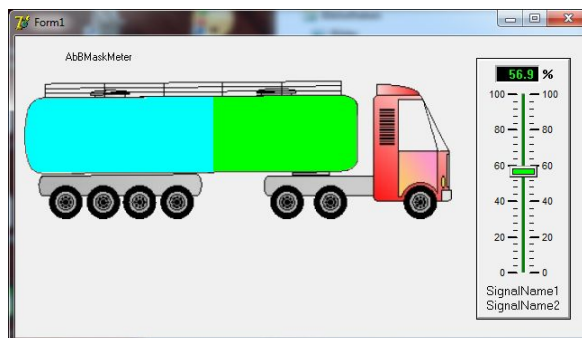
```

Result:



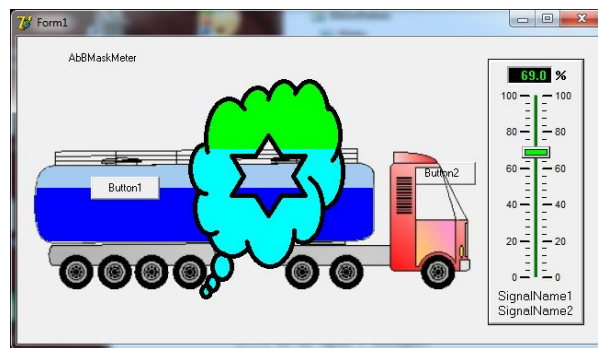
Now change the AbBMaskMeter Picture to Tankwagon.bmp

Run the application and move the slider.... OK.... not realistic... to fill a tank from left to right....



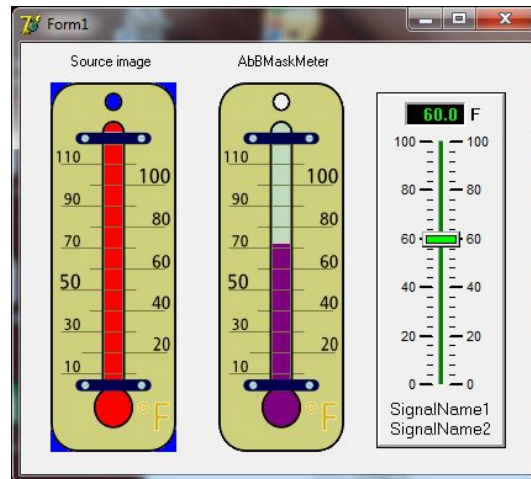
If you delete the OnChange-Event from the AbBMaskMeter it will fill the Tank more realistic...

You can place other components on top or back.... (components derived from TGraphicControl can be placed only in the background)



Info: Also ColBackground, ColForeground can be changed flicker-free...

## Scaling:



On the image above you can see a Thermometer coupled with a Slider. On the slider we have 60 deg F, on the Thermometer more than 70...

The reason for that is, that the scale (Value Range) of the TAbBMaskMeter is adjusted by default from 0..100. If you look to the clRed part of the source image the last red pixel on the bottom is at Scale position -14 and the topmost red pixel is on Scale position 129. (OK.... it's not easy to see.... you have to estimate and then fine tune... )

Now set the ValueFrom property to -14 and the ValueTo property to 129. Run the application...

