TRGBCurves v. 3.1 Shareware Component for Delphi © 2003-2010 Francesco Savastano

Supported Platforms: D4-D5-D6-D7-D2K (incl. D2010)
compatible with any version of ImageEn

Author: Francesco Savastano

Email: nws@centurybyte.com
Web : www.nwscomps.com

General Description

TRGBCurves is a visual component that allows the creation, displaying and manipulation of tonal curves for image processing. By defining a curve for each Color channel and the Luminance channel of a Full Color picture, you will be able to define a correction for each tone of the channel range (0..255), therefore applying different amounts of change for each individual tone. Curves correction is a feature found in professional image and photo editors like Adobe Photoshop.

The component can also apply the curves correction to both TBitmap and TIEBitmap encapsulated pictures.
*TIEBitmap is the class used by the ImageEn graphics library.
You can download a free copy of ImageEn at www.hicomponents.com

To download the source code of ImageEn you will need to create an account on their website and log-in in the downloads section.

or if you prefer not to use the imageEn library, you will still be able to apply the changes to any standard TBitmap (defined in the delphi graphics unit)
In this case you will need to disable the {$DEFINE USEIMAGEEN} compiler definition in the **RGBCurves.inc** file

Features

TRGBCurves allows the creation of curves for color and exposure correction in a very easy way. If you are using the ImageEn library you can preview the changes and apply the curves even directly on a TImageenView Component. Very little coding is required from your side and the entire process of previewing and applying changes is handled automatically.

- Edits each color channel and the luminance channel separately. Supports 8bits grayscale and full-color 24 bits TIEbitmap and TBitmap.
- You can add/ remove / move the points that define the curve, by the mouse interface or by code.

- A OnChangeCurve event notifies whenever something about the curves has changed
- Can Save the Curves to a text file or to a Ini File and Reload them.
- Curves may also be defined by mathematical formulas
- In addition to displaying curves and applying the image processing algorithm the component is able to get the histogram from any image and graph it together with the curves.
- Registering a TRGBCurves with a TImageenview component for previewing is also a very important feature. It allows real time preview of the curve algorithm impacting tonal changes of the pictures displayed in the TImageenView component.
- It Integrates perfectly with ImageEn Layers and Selections as well.

Trial version

Since TRGBCurves is compiled together with the ImageEn library, in order to recompile our trial dcu files on your platform you will need to have the same version of ImageEn installed.

We typically track when new imageEn versions are out and test the component when important changes are made to it. TRGBCurves is granted to be working with any imageEn version greater than 2.x.

Should any compatibility issue arise we will fix it in a short time, or release version dependent patches.

If the trial version comes with a package you can open the package, compile it and install it, otherwise.

1 Copy all dcu files and the dcr file of the component into a folder that is included in your Delphi library path.

2 Choose from Delphi menu: Component > Install Component > and select the file: NWSComps_RGBCurves.dcu

Full Version

1 First uninstall the trial component, by removing its package.

2 Copy the Package Files and the other source code files into a folder that is included in your Delphi library path.

3 Open the Package in Delphi, compile it and install it

GDIPlus Patch Information

IMPORTANT (for version 3.1):

In order to properly use TRGBCurves v. 3.1 with gdiplus you will need:

1) To enable use of the imageEn library by setting the {$DEFINE USEIMAGEEN} definition in the NWSComps_RGBCurves.inc file.

By default this definition is already ON

2) Set the value of the UseGDIPlus property of TRGBCurves to TRUE (If False normal gdi will be used)

3) <<<VERY IMPORTANT>>> If you need a correctly antialiased (smooth) curve drawing you will also need to apply the following small patch to ImageEn the iegdiplus.pas unit. It is possible that later versions of IamgeEn will include this patch , because I have contacted ImageEn's author and he's willing to add the patch in next releases. (the patch is for the current version of imageEn 3.1.2 and earlier versions). After patching, make sure that you have {$DEFINE USEGDIPLUSPATCH} in the NWSComps_RGBCurves.inc file. By default this definition is ON. Recompile imageEn package first and then the RGBCurves package. If you choose not to apply the patch then you have to disable {$DEFINE USEGDIPLUSPATCH} in the NWSComps_RGBCurves.inc file.

---

a) add the two following types to the Interface section of the iegdiplus.pas unit:

```
//Francesco Savastano --START PATCH

  TIECanvasDoublePoint = record

    X: double;

    Y: double;

  end;

  TIECanvasDoublePointArray = array of TIECanvasDoublePoint;

  //Francesco Savastano --END PATCH
```

b) add the following public method to the TIECanvas class

```
Public

//Francesco Savastano --START PATCH

procedure DrawLinesPath(thePoints: TIECanvasDoublePointArray);

//Francesco Savastano --END PATCH
```

```
...
implementation
...
//Francesco Savastano --START PATCH
procedure TIECanvas.DrawLinesPath(thePoints: TIECanvasDoublePointArray);
var
  Path: pointer;
  i: integer;
begin
  IE_GdipCreatePath(0, Path);
  try
    for i := 0 to high(thePoints) - 1 do
    begin
      IE_GdipAddPathLine(Path, thePoints[i].X, thePoints[i].Y,
                              thePoints[i+1].X, thePoints[i+1].Y);
    end;
    IE_GdipDrawPath(FGraphics,FPen.FGHandle, Path);
  finally
    IE_GdipDeletePath(Path);
  end;
end;
----------------------------------------------------------------------------------------------------
```

Included Files

NWSComps_RGBCurves.pas	
Contains the main class derived from TCustomControl	

NWSComps_RGBCurves_Types.pas	
Contains Types definitions	
NWSComps_RGBCurves_Math.pas	
Defines the ,mathematics needed to build the curves	
NWSComps_IEUtils_Previews.pas	
Defines the ,classes needed to perform real time preview on a TImageEnView component	
NWSComps_RGBCurves.inc	
Contains Compiler Directives Definitions	

********************NWSComps_RGBCurves_Types*********************

*************Types

TRGBCurves_Layout = class(TPersistent) //This class is not described here because it is used by the TRGBCurves Component in its properties and described later

TRGBCurves_CanMoveType = (cm_XY, cm_X_only, cm_Y_only, cm_CannotMove);

TRGBCurves_Channel_ViewMode = (fvmAll, fvmRed, fvmGreen, fvmBlue, fvmRGB);

TRGBCurves_FormulaCurve = function(x: single): single of object;

TRGBCurves_Editmode = (emNil, emMovePoint, emAddPoint);

TRGBCurves_Histogram = record

```
  active: boolean;
  DisplayMode: TRGBCurves_HistogramDisplayMode;
  ScaleMode: TRGBCurves_HistogramScaleMode;
  RGBMode: TRGBCurves_HistogramRGBMode;
  Reds: array[0..255] of cardinal;
  Greens: array[0..255] of cardinal;
  Blues: array[0..255] of cardinal;
  Grays: array[0..255] of cardinal;
  peakred, peakgreen, peakblue, peakgray: cardinal;
  avgred, avggreen, avgblue, avggray: cardinal;
 end;
```

TRGBCurves_HistogramDisplayMode = (HDmFilled, HDmTransparent, HDmLines);

TRGBCurves_HistogramRGBMode = (HmRed, HmGreen, HmBlue, HmGray, HmAll);

TRGBCurves_HistogramScaleMode = (HSmLinear, HSmLog, HSmauto);

TRGBCurves_LUT = array[0..255] of byte;

TRGBCurves_Lutarray = array[0..3] of TRGBCurves_LUT;

TRGBCurves_Mode = (cmFormulaCurves, cmBuildCurves, cmNoCurves);
//cmNoCurves is used for viewing just the histogram

TRGBCurves_MovePointEvent =
```
procedure(sender: TObject;
           CurrentPoint: TPoint;
          CurrentPoint_Idx: integer;
          var CanMove: TRGBCurves_CanMoveType) of object;
```

TRGBCurves_Options =
```
Set of  (Opt_Allow_AddPoints,
         Opt_Allow_RemovePoints,
         Opt_Allow_VinculatedPoints);
```

TRGBCurves_RGBmode = (cmAll, cmRed, cmGreen, cmBlue, cmRGB);

TRGBCurves_RGBSpace = (csRGB, csLum, csMixed);

TRGBCurves_Row = array[0..30000] of byte;
tpRGBCurves_Row = ^TRGBCurves_row;

********************NWSComps_RGBCurves*********************

*****************Class Types:**

Class ***TRGBCurves*** is a descendant of TCustomControl

**************TRGBCurves*************

Public / Published Properties

General

<table>
<tr><td>**CurveMode**</td><td>TRGBCurves_Mode</td><td></td></tr>
<tr><td colspan="3">This property can have 2 values:

• cmFormulaCurves
• cmBuildCurves

If you set it to cmFormulaCurves the component will show mathematical functions that you assign using the FormulaCurve_Red, …, FormulaCurve_Lum properties.

If you set it to cmBuildCurves the component will use interpolation to fit the points you provide by the interface in a very similar way to the Photoshop tonal curves.</td></tr>
<tr><td>**RGBMode**</td><td>TRGBCurves_RGBMode</td><td></td></tr>
<tr><td colspan="3">This property can have 4 values:
• cmAll
• cmRed
• cmGreen
• cmBlue

If you set RGBMode to any channel, you will see only the selected channel function/ curve.
If the mode is cmAll you will see all the functions if you are in cmFormulaCurves curve mode or you will see the luminance channel if you are in cmBuildCurves curve mode.</td></tr>
<tr><td>**RGBSpace**</td><td>TRGBCurves_RGBSpace</td><td></td></tr>
<tr><td colspan="3">This property can have 3 values:
• csRGB
• csLum
• csMixed

In RGB space, the Luminance will not be evaluated as a separate channel but split on the 3 color channels individually
In Lum space a value of Luminance is pre-calculated and treated as it were a separate channel
In Mixed space a mix of the above two spaces is used. The mix amount is established by the ColorSpaceMixAmount property</td></tr>
<tr><td>**ColorSpaceMixAmount**</td><td>integer</td><td>0-100</td></tr>
<tr><td colspan="3">Determines the amount of mixing between RGB Space and Lum Space</td></tr>
</table>

ExportDirectory	string	
Default Folder's Name that shall be displayed in the Save Dialog when the user exports the curves to a file.		
UseGDIPlus	Boolean	
If true the component will try to use GDIPlus to draw the graph elements. Requires ImageEn to work.		
UseFastPreview	Boolean	
It enables / disables fast preview when working with TImageEnView. Use this if you are dealing with big pictures. It will calculate a less-accurate preview of the current changes, speeding up the preview process.		
FastPreviewQuality	TRGBCurves_FastPreviewQuality	
Can be set to • Slow • Normal • Fast Generally the faster the preview the less the quality.		
Histogram		
HistogramDisplayMode	TRGBCurves_HistogramDisplayMode	
This property can have 3 values: • HdmFilled • HdmTransparent • HDmLines It specifies the style of the Histogram		
HistogramRGBMode	TRGBCurves_HistogramRGBMode	
This property can have 5 values: • HmRed • HmGreen • HmBlue • HmGray • HmAll The first 3 values represent the histogram for each color channel, the HmGray value represents the luminance channel and the HmAll will show all channels alltogether.		

HistogramRGBMode	TRGBCurves_HistogramRGBMode	
This property can have 5 values: • HmRed • HmGreen • HmBlue • HmGray • HmAll The first 3 values represent the histogram for each color channel, the HmGray value represents the luminance channel and the HmAll will show all channels alltogether.		
ShowHistogram	Boolean	
Default Value is FALSE. To show an histogram, you need to create one from a Tbitmap or a TIEBitmap using the **GetHistogramfromBMP** or **GetHistogramfromIEBMP** methods		
Graph Layout (TRGBCurves_Layout)		
GraphBackColor	TColor	
You can choose the color of the Graph Background. Default is White.		
BorderPercent	single	(0-100)
It's the size of the border around the Graph Area, in percent. IT changes the BorderFixed property as well. You can adjust it to enable more room for the graph labels.		
BorderFixed	Cardinal	
It's the size of the border around the Graph Area. You can adjust it to enable more room for the graph labels.		
CurveColor_Lum	TColor	
You can choose the color of the Luminance curve. Default is Black.		
CurveColor_Red	TColor	
You can choose the color of the Red curve. Default is Pure Red.		
CurveColor_Green	TColor	
You can choose the color of the Green curve. Default is Pure Green.		
CurveColor_Blue	TColor	

You can choose the color of the Blue curve. Default is Pure Blue.		
GridColor	TColor	
You can choose the color of the Grid Lines.		
GridMedianLine_Color	TColor	
You can choose the color of the median line of the grid.		
LineSize	Cardinal	
It's the size of the Line of the curve		
LineOpacity	byte	0-255
Only works when GDIPlus is used (UseGDIPlus = TRUE). You can set the amount of color opacity of the Line of a curve		
PointSize	Cardinal	
It's the size of the control Point of the curve when **Curve Mode =cmBuildCurves**		
PointOpacity	byte	0-255
Only works when **Curve Mode =cmBuildCurves** and GDIPlus is used (UseGDIPlus = TRUE). You can set the amount of color opacity of the inner part of a control point of a curve		
PointFillStyle	TBrushstyle	
Only works when **Curve Mode =cmBuildCurves** . You can set the style of the inner filling of a control point of a curve		
ShowVerticalCaptions	Boolean	
You can hide/show the labels for the Y axis of the graph. This is useful if you want to display an histogram without the curves, since the histogram has no vertical labels.		
ShowMedianline	Boolean	
Default is TRUE. The median dash-dotted line is displayed to visualize the discard of the curve from the no-change position.		
Font	Tfont	

Choose the font to display text in the graph.		
Curve Behaviour		
Points	TRGBCurves_DoublePointsarray	
Contains the array of Points that make-up the curve that is currently displayed Pertains to **Curve Mode = cmBuildCurves**		
RedPoints	TRGBCurves_DoublePointsarray	
Contains the arrays of Points for the RED channel Pertains to **Curve Mode = cmBuildCurves**		
GreenPoints	TRGBCurves_DoublePointsarray	
Contains the arrays of Points for the GREEN channel Pertains to **Curve Mode = cmBuildCurves**		
BluePoints	TRGBCurves_DoublePointsarray	
Contains the arrays of Points for the BLUE channel Pertains to **Curve Mode = cmBuildCurves**		
AllPoints	TRGBCurves_DoublePointsarray	
Contains the arrays of Points for the LUMINANCE channel Pertains to **Curve Mode = cmBuildCurves**		
MinFittingX	Integer	***Read Only***
InterpLinear_Min	integer	Default is 15.
Changes the least amount of local linearization that should be applied to the interpolated curve. Local linearization makes the curve shape more stable, however it tends to segment the curve.		
InterpLinear_Max	integer	Default is 65.
Changes the maximum amount of local linearization that can be applied to the interpolated curve.Local linearization makes the curve shape more stable, however it tends to segment the curve.		
First X Coordinate for which the Curve value is Minimum (Y=0).		
MaxFittingX	Integer	***Read***

		Only
First X Coordinate for which the Curve value is Maximum (Y=255).		
FormulaCurve_Red	TRGBCurves_FormulaCurve	
Assign to this property the function that you want to use for the RED channel when **Curve Mode = cmFormulaCurves**		
FormulaCurve_Green	TRGBCurves_FormulaCurve	
Assign to this property the function that you want to use for the GREEN channel when **Curve Mode = cmFormulaCurves**		
FormulaCurve_Blue	TRGBCurves_FormulaCurve	
Assign to this property the function that you want to use for the BLUE channel when **Curve Mode = cmFormulaCurves**		
FormulaCurve_Lum	TRGBCurves_FormulaCurve	
Assign to this property the function that you want to use for the LUMINANCE channel when **Curve Mode = cmFormulaCurves**		

Public / Published Methods

ApplyCurvesToBitmap		*Proc*
params • thebitmap : Tbitmap ************ Apply the Curves that were defined by mathematical formulas or Built by control points to a TBitmap object. Pixel format must be supported.		
ApplyCurvesToIEBitmap		*Proc*
params • theiIEbmp : TIEBitmap • Mask: TIEMask; • EditedRect: TRect; • const bUseMask: boolean		

****

Apply the Curves that were defined by mathematical formulas or Built by control points to a TIEBitmap object.

The Mask is a ImageEn Selection that defines any area inside the TIEBitmap in which the changes should be applied.

The EditedRect is the rectangular crop area in which the changes should be applied

bUseMask defines whether to consider the Mask or ignore it.

Pixel format must be supported.

ApplyCurvesToImageenView *Proc*

params
- theieview : TimageenView

****

Apply the Curves that were defined by mathematical formulas or Built by control points to a TImageEnView of choice. Current layer will be affected. If a selection (TIEMask) is defined only selected parts will be affected by changes.

Pixel format must be supported.

IEView_Preview_Register *Proc*

params
- theieview : TimageenView

****

Registering a TImageenView component to work in conjunction with TRGBCurves will automatically reflect any changes made to the curves on the registered TImageEnView display. The preview will affect the current layer and will also take care of any selection (if present). Every changes in the display (zooming, panning, selecting layers, creating selections) will instantaneously update the preview.

IEView_Preview_UnRegister *Proc*

params
- NONE

****

After a TimageenView component has been registered for previews by the **IEView_Preview_Register** method, you will need to unregister it by this command. Unregistering the component will stop TRGBCurves from updating the TImageenview display.

IEView_Preview_ApplyChanges *Proc*

params

<table>
<tr><td colspan="3">•	theieview : TimageenView

****
Apply the Curves that were defined by mathematical formulas or Built by control points to the TImageEnView that was registered for previews. The last changes previewed will be applied on the current layer and If a selection (TIEMask) is defined only selected parts will be affected by changes.

Pixel format must be supported.</td></tr>
<tr><td><b>GetHistogramfromBMP</b></td><td></td><td><i>Proc</i></td></tr>
<tr><td colspan="3">**params**
•	thebitmap : Tbitmap

****
Loads the histogram from a Tbitmap object.</td></tr>
<tr><td><b>GetHistogramfromIEBMP</b></td><td></td><td><i>Proc</i></td></tr>
<tr><td colspan="3">**params**
•	theiIEbmp : TIEBitmap

****
Loads the histogram from a TIEBitmap object.</td></tr>
<tr><td><b>GetHistogramfromImageEnView</b></td><td></td><td><i>Proc</i></td></tr>
<tr><td colspan="3">**params**
•	theieview : TimageenView

****
Loads the histogram from a TimageenView object.</td></tr>
<tr><td><b>ExportCurves</b></td><td></td><td><i>Proc</i></td></tr>
<tr><td colspan="3">**params**
•	NONE
****
Opens the save dialog to export curves to text files.</td></tr>
<tr><td><b>ImportCurves</b></td><td>Boolean</td><td><i>Func</i></td></tr>
<tr><td colspan="3">**params**
•	NONE
****
Opens the open dialog to import curves from text files. Returns True if any curve is imported.</td></tr>
<tr><td><b>ExportCurvesToINI</b></td><td></td><td><i>Proc</i></td></tr>
<tr><td colspan="3">**params**
•	Inifile: Tinifile</td></tr>
</table>

• IniSection: string ************ Exports curves to INI files.		
ImportCurvesFromINI		*Proc*
params • Inifile: Tinifile • IniSection: string ************ Imports curves from INI files.		
ResetPoints		*Proc*
params NONE ************ Reset all curves by resetting the values of all the control points to default. Only works with **Curve Mode = cmBuildCurves**		
SetthePoint		*Proc*
params • ix: integer • thepoint: TRGBCurves_doublepoint ************ Set the value of the point at the position ix of the current curve Only works with **Curve Mode = cmBuildCurves**		
SetCurrentPoint		*Proc*
params • thepoint: TRGBCurves_doublepoint ************ Set the value of the current point (the one that was last moved or created) of the current curve Only works with **Curve Mode = cmBuildCurves**		
GetCurrentPoint	TRGBCurves_doublepoint	*Func*
Get the current point (the one that was last moved or created) of the current curve Only works with **Curve Mode = cmBuildCurves**		
AddPoint		*Proc*
params • thepoint: TRGBCurves_doublepoint ************ Add a point to the curve Only works with **Curve Mode = cmBuildCurves**		
RemovePoint		*Proc*

<table>
<tr><td colspan="3">**params**
• ix: integer
****
Remove the point at the position ix of the current curve
Only works with <b>Curve Mode = cmBuildCurves</b></td></tr>
<tr><td><b>RemovePoint_Current</b></td><td></td><td><i>Proc</i></td></tr>
<tr><td colspan="3">**params**
• NONE
****
Remove the current point (the one that was last moved or created) of the current curve
Only works with <b>Curve Mode = cmBuildCurves</b></td></tr>
<tr><td><b>PointExists</b></td><td>boolean</td><td><i>Func</i></td></tr>
<tr><td colspan="3">**params**
• const thepoint: TRGBCurves_doublepoint
****
Retunrs TRUE if the specified point already exists in the current curve.
Only works with <b>Curve Mode = cmBuildCurves</b></td></tr>
<tr><td><b>ExportLUT</b></td><td><b>TRGBCurves_LUT</b></td><td><i>Func</i></td></tr>
<tr><td colspan="3">**params**
• mode: TRGBCurves_RGBMode
****
The result of this function is a Lookup Table. It contains the ordered curve values for each of the [0--255] tones of a channel. The channel is specified by the Mode parameter.</td></tr>
<tr><td><b>ExportLUTs</b></td><td></td><td><i>Proc</i></td></tr>
<tr><td colspan="3">**params**
• var BGRALUTarray: TRGBCurves_Lutarray
• var GrayLUT: TRGBCurves_LUT
****
The returned parameters contain the Lookup Tables for all channels. GrayLut is the luminance channel.</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

Public Events

<table>
<tr><td><b>OnChangeCurve</b></td><td>TNotifyEvent</td><td></td></tr>
<tr><td colspan="3">Use this event to be notified whenever a change happens to the curves.</td></tr>
<tr><td><b>OnChangeRGBMode</b></td><td></td><td></td></tr>
<tr><td colspan="3">Use this event to be notified whenever the RGBMode changes</td></tr>
</table>