TMS GUIMotions
DEVELOPERS GUIDE

Dec 2016

Web: http://www.tmssoftware.com
Email: info@tmssoftware.com

Index

TMS GUIMotions availability

TMS GUIMotions is available as VCL component for Win32/Win64 application development.

TMS GUIMotions is available for Delphi 7, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10 Seattle, 10.1 Berlin (Prof/Enterprise/Architect)

TMS GUIMotions description

TMS GUIMotions is a component designed to display and animate images, custom drawn information or controls in various spectacular ways.

TMS GUIMotions important note

TMS GUIMotions uses the Microsoft DirectX 9 runtime for its fluent 3D image animations. The Microsoft DirectX 9 runtime is preinstalled on Windows XP and Windows Vista systems. For older operating systems, the Microsoft DirectX runtime can be separately downloaded from Microsoft. TMS GUIMotions is currently compiled for DirectX 9.0C with the DLL d3d9x_33.dll. It is recommended that either the presence of this DLL is checked or is installed in the \Windows\System32 directory before compilation / installation of the component as well as for deployment with applications. When the installer of the TMS GUIMotions component is used, this file is copied automatically to \Windows\System32.

TMS GUIMotions always tries to render on the video hardware of your pc. When a graphical card with 3D hardware acceleration is found, the component will check whether it can be used with multi-sampling (alpha blending and anti-aliasing). If it fails the component will try to create a 3D device with software rendering. This depends on the 3D graphical card you are using.

Note that on some systems, DirectX 9 might be installed by Direct3D not enabled. This can be verified by running the tool dxdiag.

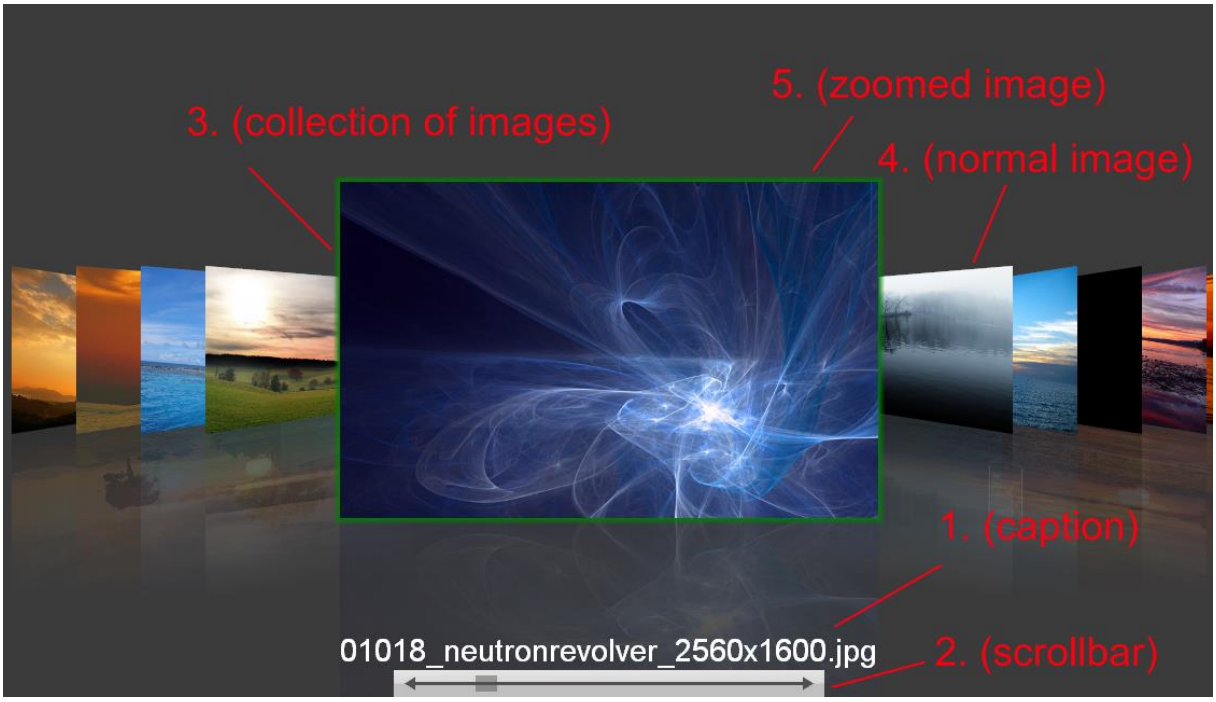
For troubleshooting:

- Run DxDiag and make sure Direct3D is enabled and working (test button).
- Install the latest drivers of your 3D graphical card.
- Verify if you have the latest version of DirectX 9 installed.

TMS GUIMotions visual organisation

The TMS GUIMotions component has 8 modes to display & animate your images, custom drawn information or controls. Animation can be controlled with the mouse or the keyboard. In the screenshot below is an overview of the major and important elements of the component.

Overview major elements



1) When selecting an image the caption[1] will have the caption text assigned to the image. The caption can be positioned in every corner or in the top or bottom center of the control.

2) The scrollbar[2] can be used to navigate through the collection of images. This scrollbar can be positioned in every corner or in the top or bottom center of the control.

3) The most important element of the control is the collection of images[3]. The images in this collection will eventually be loaded in the DirectX video memory. DirectX will create a texture on a vertex buffer which can be animated. You can define these images on three different ways:

    a. Normal image: just add a PNG, JPEG, BMP, GIF file to the collection. Note that the formats supported depend on the installed TPicture formats.
    b. Custom painted image : paint on the canvas of the custom image (OnImageCustomDraw required) and this will be animated
    c. Control painted image : Paint a control on the image that will be animated

    For each image you can also add an alternate image which can be created on the same way as the normal image (see the three image types above). The alternate image can be optionally shown when right-clicking on the selected image.

4) Depending on the animation mode choosen, the normal images will be loaded skewed, scaled or transformed. When you select an image the selected image will transform to a different state.

5) When you click on the selected image you can zoom in depending on the properties PictureWidthZoomed and PictureHeightZoomed you have chosen.

There are three ways of loading images into video memory (controlled by the ImageQuality property)

a) iqNone : lower resolution for minimum memory usage and maximum animation speed (PictureWidth and PictureHeight properties will be used to set resolution to load Image).
b) iqSelected : Only the selected image will be loaded in high resolution (for reasonable animation speed and memory usage)
c) IqFull: all images will be loaded in video memory at full resolution. (longer loading time for images and higher memory usage)

Animation modes

Carousel:

The carousel mode animates the images from left to right with a skew on the non-selected images. On the selected image you can click to zoom in and when zoomed in you can right-click to display the alternate image (back of the image) with a page flip animation.

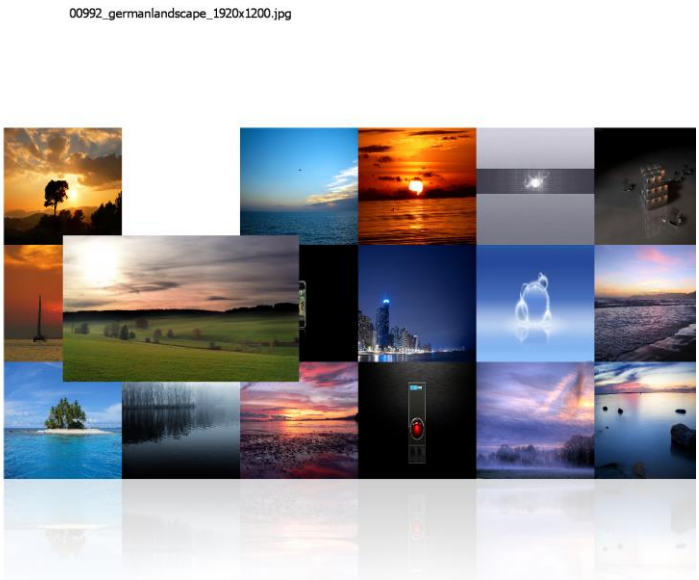
Imageband left/right:

In the Imageband left / right mode the images go from left to right or vice versa depending on the mode you have chosen. If you click on a selected image the image goes to the center of the control and the other images go to the left or right side. A second click on the selected image will zoom the image and a right-click can optionally flip it to the alternate image.

Conical inner / outer band:

The conical inner / outer band animates its images in a cylinder. In this mode you can zoom in or out and right click to animate to the alternate image.

Imagestrip:



The Imagestrip mode animates its images in a strip. You can choose how many rows the Imagestrip displays as well as the size of the images in the strip (PictureWidth/PictureHeight). Also in this mode you can select the alternate image and zoom in or out.

Book / Rolodex:

The book and the rolodex modes are stacked images and can be animated with the keyboard or with the mouse, you can drag & drop. Each time you change a picture, it animates to display a page turn effect.

TMS GUIMotions programmatic use

As explained in the previous chapter, TMS GUIMotions can display image files, custom drawn information or controls. In this chapter, it will be shown how either images, custom drawn information or controls can be added to the component.

Initializing images

There is not much code needed to initialize and start using the component. First of all, images need to be added to the image collection, either in design time or in code.

Directly loading the pictures in code:

```
for I := 1 to 12 do
begin
  with GUIMotions1.Images.Add do
  begin
  //Add picture already loaded
  Picture.LoadFromFile('image1.jpg');
  //Same for adding alternate picture but specify AlternatePicture
  //Set some properties…
  end;
end;
```

Letting GUIMotions load the image from file by just setting the filename:

```
for I := 1 to 12 do
begin
  with GUIMotions1.Images.Add do
  begin
  //Just set reference to the image filename, GUIMotions will load the
  //image when needed
  PictureLocation := 'image1.jpg';
  //Same for adding alternate picture but specify AlternatePicture
  //Set some properties…
  end;
end;
```

Letting GUIMotions load the image in a separate thread from file by just setting the filename:

```
for I := 1 to 12 do
begin
  with GUIMotions1.Images.Add do
  begin
  //Just set reference to the image filename, GUIMotions will load the
  //image when needed
  ThreadPictureLocation := 'image1.jpg';
  end;
end;
```

Load many images at once:

```
GUIMotions1.AddImagesFromFolder('.\Images\*.jpg');

GUIMotions1.AddImageLocationsFromFolder('.\Images\*.jpg');
(or any other format supported by TPicture)

GUIMotions1.AddThreadedImageLocationsFromFolder('.\Images\*.jpg');
(images are loaded in a separate thread)
```

Initializing controls

The TMS GUIMotions control can also be used to display and paint controls that have the ability to draw to a canvas (ie. properly implement the PaintTo() method). To use controls, first references to the controls need to be added to the Images collection. Secondly, controls need to be set visible and positioned correct at the appropriate events. Here is a small tutorial which might help to setup similar projects.

Step 1: adding references to the controls to the collection. (Sample with chart control)

```
for I := 0 to 10 do
begin
  with GUIMotions1.Images.Add do
  begin
    // 1) Create control with parent GUIMotions1
    c := TAdvGDIPChartview.Create(GUIMotions1);
    // 2) Set the control visible to false
    c.Visible := false;
    // 3) Set the parent to GUIMotions1
    c.Parent := GUIMotions1
    // 4) Add some random data to your control
    // 5) Add control to image collection
    Control := c;
  end;
end;
```

This will by default only show white images (as nothing is loaded in the picture property). In case a GUIMotions.DefaultPicture is set the default picture will be used.

Step 2: making sure the control is displayed in the animation:

Add an event handler for GUIMotions.OnImageCustomDraw. This event can be called when the GUIMotions component is trying to create a texture. When no picture was set for the image from the collection to be rendered, GUIMotions will call this custom drawing event. In this event you need to draw the control on the canvas of the picture variable.

```
procedure TForm42.GUIMotions1ImageCustomDraw(Sender: TObject; Image:
Integer; Picture: TBitmap);
var
  c: TAdvGDIPChartView;
begin
  c := TAdvGDIPChartView(GUIMotions1.Images[Image].Control);

  c.Width := GUIMotions1.PictureWidthZoomed;
  c.Height := GUIMotions1.PictureHeightZoomed;

  picture.Width := GUIMotions1.PictureWidthZoomed;
  picture.Height := GUIMotions1.PictureHeightZoomed;

  c.PaintTo(Picture.Canvas, 0, 0);
```

```
  Picture.Canvas.Pen.Color := clBlack;
  Picture.Canvas.Pen.Width := 2;
  Picture.Canvas.Brush.Style := bsClear;
  Picture.Canvas.Rectangle(0,0,Picture.Width,Picture.Height);

end;
```

Step 3: show and hide the control:

The control has 3 events that can be used to make the control visible. OnImageClick, OnImageDblClick, OnImageZoomed. In this case we are using the OnImageZoomed event.

```
procedure TForm42.GUIMotions1ImageZoomed(Sender: TObject; Image: Integer;
  ImageRectangle: TRect);
var
  r: TRect;
begin
  AdvChartPanesEditorDialogGDIP1.ChartView :=
TAdvGDIPChartview(GUIMotions1.Images[Image].Control);
  r := ImageRectangle;
  TAdvGDIPChartview(GUIMotions1.Images[Image].Control).SetBounds(r.Left,
r.Top, r.Right - r.Left, r.Bottom - r.Top);
  TAdvGDIPChartview(GUIMotions1.Images[Image].Control).visible := true;
end;
```

From the OnImageUnzoom, the control can be hidden again.

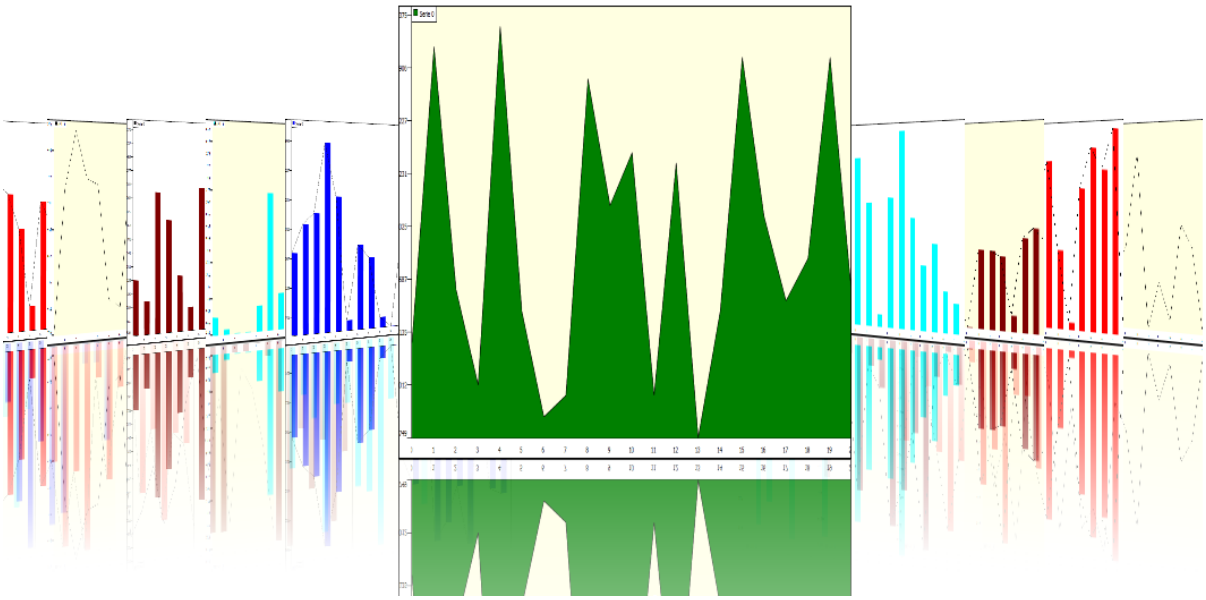
```
procedure TForm42.GUIMotions1ImageUnZoom(Sender: TObject; Image: Integer;
  ImageRectangle: TRect);
var
  r: TRect;
begin
  TAdvGDIPChartview(GUIMotions1.Images[Image].Control).visible := false;
end;
```

Step 4: Update the image / texture upon control changes:

Wherever you make changes to the control you need to force an update to pass the changes of the control to the image. The update will automatically call the OnImageCustomDraw event and will paint the latest updated control on the image.

```
GUIMotions1.Images[GUIMotions1.ImageSelected].Update;
```

Here is a screenshot to give you an idea what the result of this tutorial can be:



TMS GUIMotions important methods and properties

Events

TMS GUIMotions control contains some important events to interact with, after or during the animation.

To add custom code when a picture is clicked, add the event OnImageClick (for the alternate picture, add OnImageAlternateClick, OnImageDblClick when an image is double clicked). A special parameter of the OnImageClick event is AllowZooming. With this parameter you can disable zooming by setting it to false.

When an image is changed the event OnImageChanged will be called. Via parameters of the event, the current select image and the previously selected image is returned.

If no picture or alternate picture is specified in the Image of the images collection, the OnImageCustomDraw or OnImageCustomAlternateDraw event is called. You can draw on a bitmap canvas and use that bitmap as texture.

When an image is loaded and its texture is created the event OnImageLoaded is called. When an image is selected (after the animation is complete) the OnImageSelected event is called.

If you click on the image and the image is zoomed then after the animation completes the event OnImageZoomed is called. When you click again the image is starting to zoom out and then the event OnImageUnZoom is called.

Properties

Below are the properties to change the look and feel and the usability of the component.

Animationfactor: The Animationfactor is the global factor used to choose the speed in each AnimationMode. A higher factor is a slower animation speed and vice versa.

AnimationMode: There are currently 8 animation modes available. When a different AnimationMode is chosen the pictures will be loaded in a different position and shape and will be animated in the chosen mode.

BackColor: This is the background color of the component.

CaptionFont: Sets the font with which the caption text for an image in the images collection is displayed.

CaptionPosition: The position of the caption on the control.

CaptionLeft and CaptionTop: When the CaptionPosition is spCustom, the left and the top properties can be used to position the caption on the control.

Glow: Enable glow on the selected image. When an image is selected, an extra glow effect around the border of the image is displayed. Note that the glow is not available for the book and rolodex animation mode.

GlowAnimation: Animates the glow when an image is selected.

GlowColor: The color of the glow.

GlowSize: The size of the glow.

ImageLoading: You can choose from LoadAll, LoadOnDemand and LoadOnDemandAndRelease.

LoadAll: Immediately loads every image added to the collection as textures.

LoadOnDemand: Loads only visible images from the collection and when a next image is selected the control will load the next images. When returning to the previous image. The picture will remain loaded.

LoadOnDemandAndRelease: The same as LoadOnDemand but when a picture is not visible it will be released and recreated when navigating through the collection.

ImageQuality: There are 3 quality types to choose from.

None: The images will be loaded with the PictureWidth and the PictureHeight from the control. The quality will automically be reduced to this chosen size.

Selected: Upon selection, the lower resolution image will be released and reloaded with the full picture width and height and will as such be displayed with maximum resolution.

Full: Load all images at full quality.

ImageRowCount: When the AnimationMode Imagestrip is selected the number of rows is set with this property.

Images: The collection of images (when loaded in code the images will only be visible at runtime).

ImageSelected: The selected image of the collection.

KeyboardLookup: Enable KeyboardLookup to easily navigate to the image with the caption of the letters typed on the keyboard.

Mirror: When mirror effect is enabled there will be a reflection of the image.

MirrorBottomAlpha / MirrorTopAlpha: The alpha transparency value of the top and the bottom of the image reflection.

MirrorColor: The color of the overlay on the reflected image.

PictureDistance: The distance between two images in animation modes Carousel, Imageband left and right

PictureHeight / PictureWidth: The size of the animated image (the quality will be determined by the original loaded picture and if ImageQuality is set to selected or full).

PictureHeightZoomed / PictureWidthZoomed: The size of the animated image when zoomed in. When ImageQuality is set to selected or full the zoomed image will contain its full resolution. When set to iqNone the zoomed image will contain a resolution set to PictureWidth and PictureHeight (this mode will decrease the CPU load and will load the images faster).

ScrollBar: When the scrollbar is enabled you can hold and drag the slider to navigate through the collection of images. Several properties can be used to change the look of the scrollbar (color, slidercolor, arrowcolor, …).

SkewPercentage: Sets the picture skew percentage (only in mode carousel).

SlideOutDistance: The distance for the pictures to slide out when the selected picture is zoomed in.

TMS GUIMotions keyboard and mouse handling.

When running the component you can either animate / navigate with the mouse or with the keyboard. Here you have a list of the important functions when working with the control.

Mouse

Left button on image: select the image with animation
Left button not on image: zoom out when image is zoomed in.

Right button: select the alternate / normal image with animation

Mouse wheel scroll up/down: select the next previous picture.

Keyboard

Left arrow: select previous image
Right arrow: select next image
Spacebar: select alternate image with animation
Return: zoom in on selected image.
Page Up / Page Down: navigate through images selecting the next ten images.
Home: Select the first image
End: Select the last image

Keyboard lookup support

When the KeyboardLookup property is enabled type the name (or the first letters / numbers) of the image and the control will navigate to the image with that caption text.