VCL

TMS THotSpotImage
DEVELOPERS GUIDE

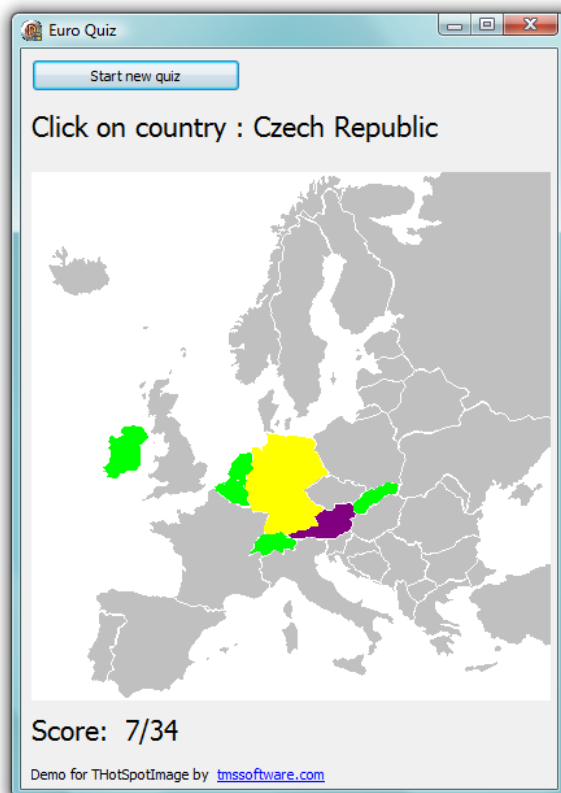## Table of contents

THotSpotImage availability

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

VCL versions:
THotSpotImage is available for Delphi 7, 2006, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, XE9, 10,10.1, 10.2, 10.3, 10.4 & C++Builder 2006, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, XE9, 10,10.1, 10.2, 10.3, 10.4.

THotSpotImage use

The TMS THotSpotImage component is designed to allow indication, selection, handling of clicks on arbitrary shapes (hotspots) within an image. Such a arbitrary shape or hotspot can be displayed in a different color or with a different superimposed image for mouse hover, click and blink state. A HotSpotEditor component is available with which hotspots can be graphically edited both at runtime and designtime.

THotSpotImage organisation

The THotSpotImage consists of an image and a collection of hotspots. The hotspot collection holds the information for each hotspot. A hotspot has a shape : rectangle, ellips or polygon. In addition it has settings for image to show in clicked & hovered state and color to use in clicked, selected, down & hovered state. Each hotspot can also have a hint. Using the THotSpotImage is as simple as adding definitions for the hotspot in the HotSpots collection. The THotSpotImage has different events that are triggered upon mouse interaction with the HotSpots.

THotSpotImage setup and methods

Properties

The specific properties available in THotSpotImage are:

THotSpotImage.AutoSize: Boolean

When true, the size of the THotSpotImage component automatically adapts to the of the picture used for the HotSpotImage.

THotSpotImage.BlinkInterval: Integer

This sets the frequency for a blinking hotspot

THotSpotImage.HotSpotCursor: TCursor

This property sets the cursor to use when the mouse is inside a hotspot.

THotSpotImage.HotSpots: THotSpots

This is the collection of the definitions of the hotspots

THotSpotImage.Images: TImageList

This is the imagelist that holds images that can be displayed over hotspots on click or hover.

THotSpotImage.Picture: TPicture

This sets the HotSpotImage background picture. This picture can be in any Delphi/C++Builder picture format. Standard this is BMP, JPEG, ICO, WMF (and GIF from Delphi 2009)

THotSpotImage.SelectionMode: TSelectionMode

This selects the SelectionMode. SelectionMode can be:

- smSingleSelect: a single hotspot can be selected at a time.
- smMultiSelect: multiple hotspots can be selected by simple mouse clicks
- smMultiSelectCtrl: Ctrl-click performs multiselect, simple click selects one hotspot

THotSpotImage.Stretch: Boolean

When true, the THotSpotImage background picture stretches to the size of the control

THotSpotImage.Transparent: Boolean

When true, the THotSpotImage background picture is displayed transparent. The transparency color value is determined by the bottom left pixel.

Events

THotSpotImage.OnHotSpotClick: triggered on a single click on a hotspot

THotSpotImage.OnHotSpotDblClick: triggered on a double click on a hotspot

THotSpotImage.OnHotSpotEnter: triggered when the mouse enters a hotspot

THotSpotImage.OnHotSpotExit: triggered when the mouse leaves a hotspot

THotSpotImage.OnHotSpotRightClick: triggered on a right click on a hotspot

HotSpots collection

This collection consists of objects of the type THotSpot. The THotSpot has following properties:

THotSpot.ShapeType: TSpotShapeType

This sets the type of the hotspot shape:
- stRectangle: hotspot shape is rectangular
- stEllips: hotspot shape is ellips
- stPolygon: hotspot shape is a polygon

THotSpot.Hint: string

Sets the hint for a hotspot

THotSpot.Name: string

Sets a name for a hotspot

THotSpot.ID: integer

Sets an ID for a hotspot

THotSpot.X: integer

Sets the X coordinate of the hotspot. For an ellips or rectangle, this is the left coordinate

THotSpot.Y: integer

Sets the Y coordinate of the hotspot. For an ellips or rectangle, this is the top coordinate

THotSpot.Width: integer

Sets the width of the hotspot for an ellips or rectangle

THotSpot.Height: integer

Sets the height of the hotspot for an ellips or rectangle

THotSpot.Clipped: Boolean

When true, the hotspot clicked, seleced, hover, blink image is displayed clipped within the shape of the hotspot

THotSpot.HoverImage: TPicture

Sets the image to display over the hotspot on mouse hovering

THotSpot.ClickImage: TPicture

Sets the image to display over the hotspot when clicked

THotSpot.SelectedImage: TPicture

Sets the image to display over the hotspot when the hotspot is in selected state

THotSpot.BlinkImage: TPicture

Sets the image to display over the hotspot when the hotspot is in blinking state

THotSpot.HoverImageIndex: integer

Sets the image from the imagelist to display over the hotspot on mouse hovering

THotSpot.ClickImageIndex: integer

Sets the image from the imagelist to display over the hotspot when clicked

THotSpot.SelectedImageIndex: integer

Sets the image from the imagelist to display over the hotspot when selected
THotSpot.BlinkImageIndex: integer

Sets the image from the imagelist to display over the hotspot when blinking

THotSpot.Angle: integer

Sets the angle of the ellips or rectangle shape

THotSpot.Down: Boolean

Defines whether a hotspot is in down state or not

THotSpot.HoverColor: TColor

Sets the background color of the hotspot on mouse hovering. The color is only used when no hover image is assigned.

THotSpot.ClickColor: TColor

Sets the background color of the hotspot when clicked. The color is only used when no click image is assigned.

THotSpot.SelectedColor: TColor

Sets the background color of the hotspot when selected. The color is only used when no selected image is assigned.

THotSpot.BlinkColor: TColor

Sets the background color of the hotspot when blinking. The color is only used when no hover image is assigned.
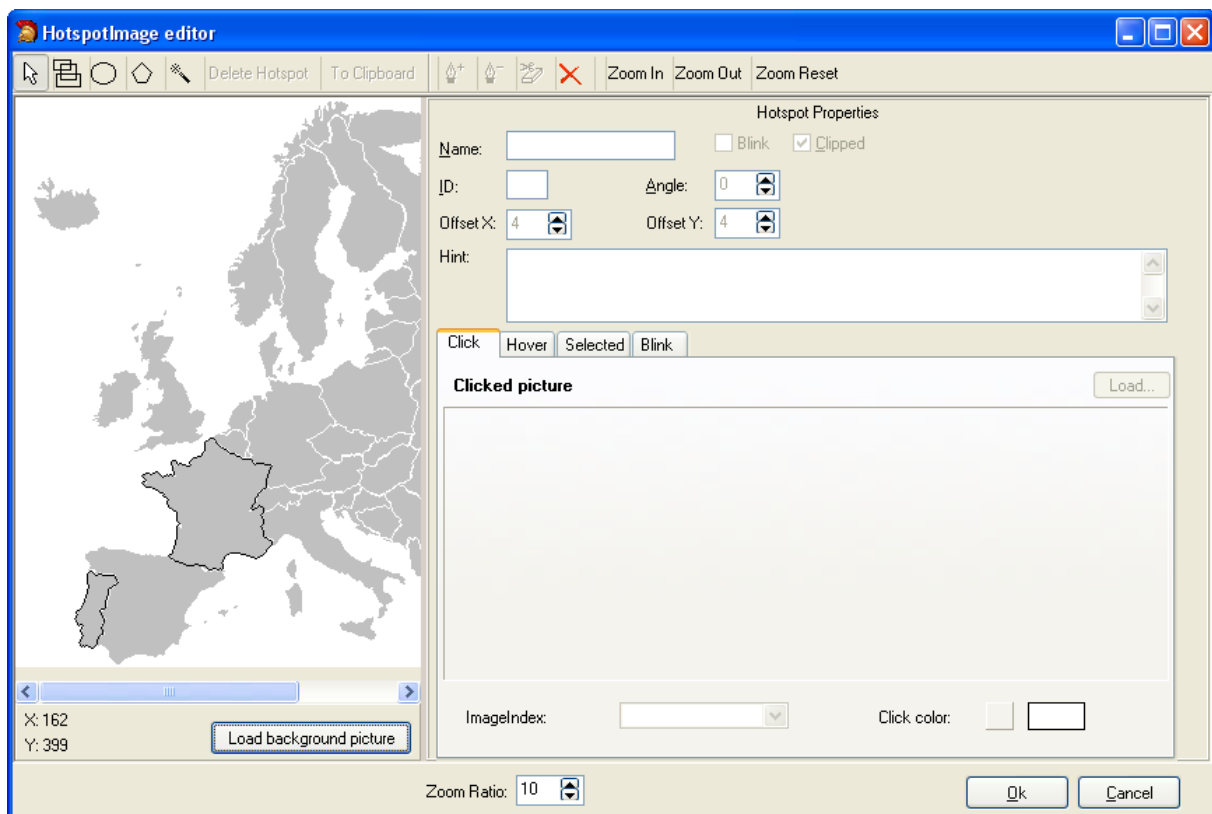
THotSpot.Blink: boolean

When true, the hotspot is in blinking state

THotSpot.Selected: Boolean

When true, the hotspot is in selected state

While the HotSpots collection can be programmatically accessed, there is also a HotSpot editor provided. This HotSpot editor is invoked when clicking the HotSpots property in the Object Inspector:

THotSpotImage getting started with the HotSpot editor

Drop a THotSpotImage on the form and load a background picture via the Picture property. Start the HotSpots editor by clicking the HotSpots property in the Object Inspector.

Select the magic wand tool to automatically select an area:
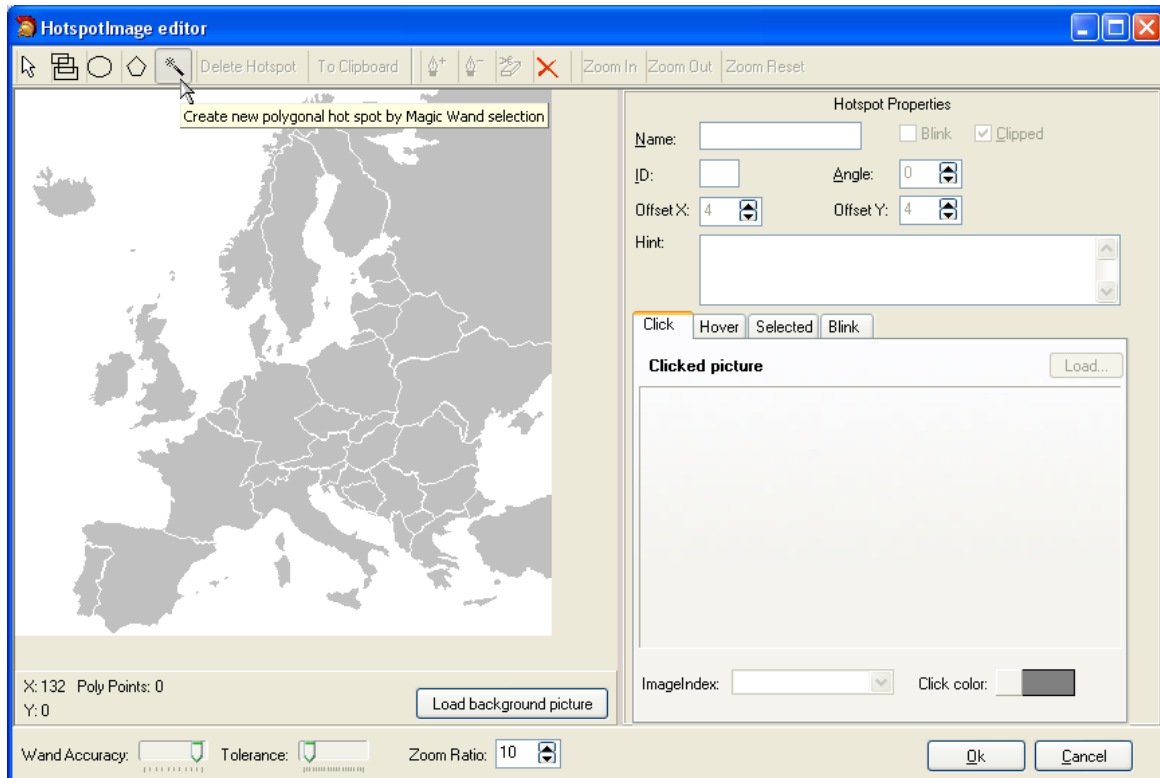
After clicking inside the area of France and going to selection mode again by clicking the first arrow button on the toolbar, this becomes:
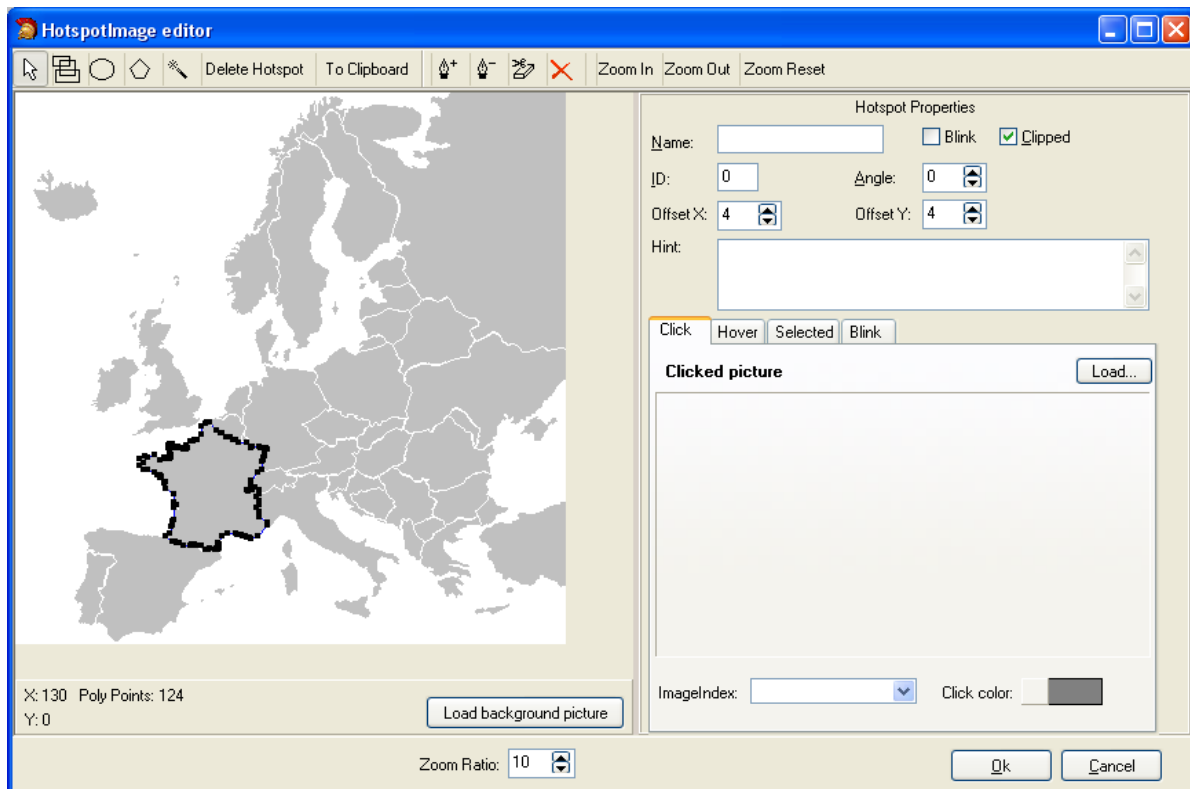
On the right pane, the properties for the hotspot can be edited. The name, ID, Hint, Angle and state of the hotspot can be set. In the different tabs, either image, color or imagelist image can be set for the different states of the hotspot. For example, moving to the Selected tab and setting the Selected color to orange results in:
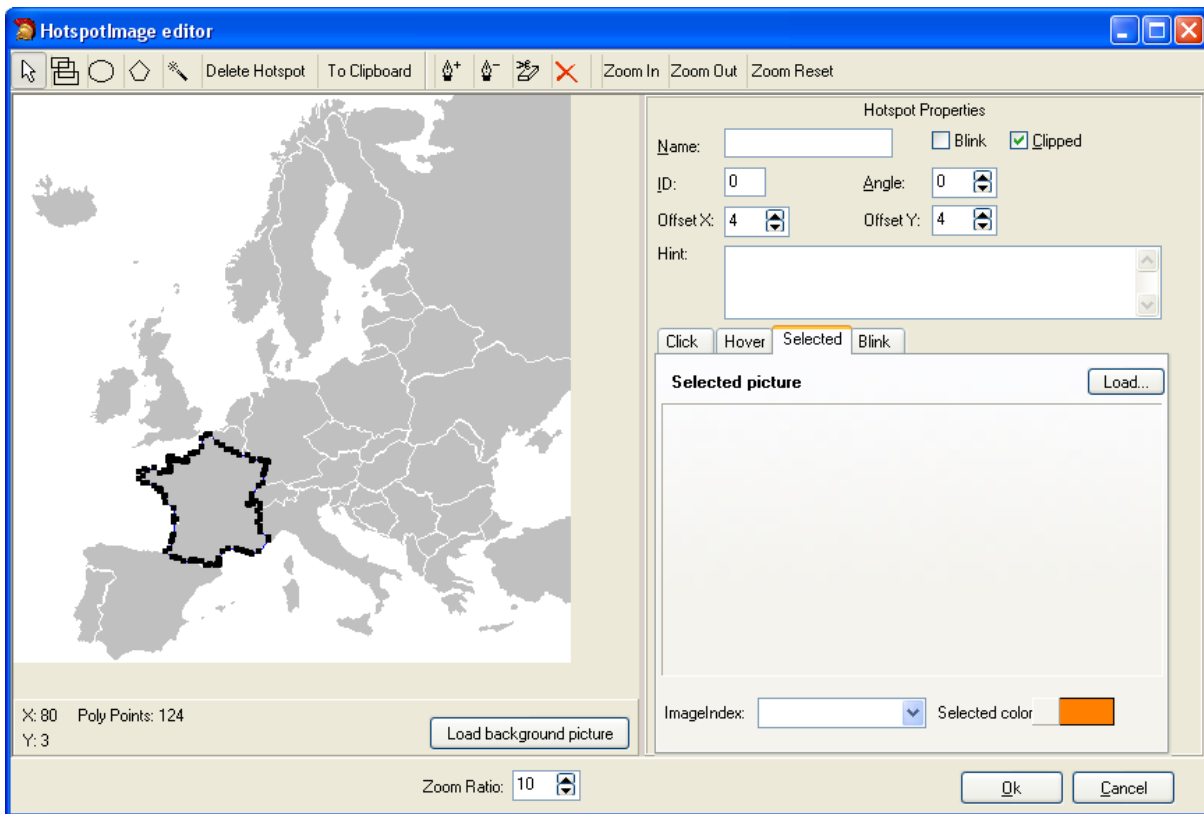
After clicking the hotspot at runtime to select it, this appears as:

THotSpotImage programmatic access to hotspots

It is equally possible to programmatically add & delete hotspots.

This code snippet adds an ellips shaped hotspot to the THotSpotImage:

```
with HotSpotImage.HotSpots.Add do
begin
  X := 50;
  Y := 50;
  Width := 75;
  Height := 75;
  ShapeType := stEllipse;
  Clipped := true;
  SelectedColor := clLime;
  Hint := 'This is a programmatically added ellipse';
  Name := 'Circle close to Iceland';
  ID := 1;
end;
```

The result after clicking with the following OnHotSpotClick event handler:

```
procedure TForm2.HotSpotImageHotSpotClick(Sender: TObject; HotSpot:
THotSpot);
begin
  ShowMessage('Hotspot clicked: '+ IntToStr(HotSpot.ID) + ':'+
HotSpot.Name);
end;
```


This results in:

THotSpotImage runtime saving & loading of HotSpots

Hotspots can be easily saved & loaded at runtime. The HotSpotImage.HotSpots collection has following methods:

THotSpotImage.HotSpots.SaveToStream(S: Stream);
THotSpotImage.HotSpots.LoadFromStream(S: Stream);
THotSpotImage.HotSpots.SaveToFile(FName: string);
THotSpotImage.HotSpots.LoadFromFile(FName: string);