

# **Greatis Form Skin**

**Form Skin and Transparency Components  
for Borland Delphi / C++ Builder**

## **User Manual**

**Copyright © 2001 Greatis Software**

Contents	1
About Greatis Form Skin	2
Classes hierarchy	2
TCustomFormSkin	2
TSimpleFormSkin	6
TBitmapFormSkin	6
Installing and uninstalling	8
License agreements	9
Contact information	10

## About Greatis Form Skin

Form Skin pack contains components that allow to create forms with transparent areas and forms with arbitrary shape:

### TCustomFormSkin

Base parent component for all skin components. It defines all methods, properties and events needed to create skinned forms.

### TSimpleFormSkin

This component just publish some properties and events, derived from TCustomFormSkin.

### TBitmapFormSkin

This component allows to create form skin by bitmap. Just draw your form in any graphic editor, load it into Skin property and set transparent color.

## Classes hierarchy

```

TObject
  TPersistent
    TComponent
      TCustomFormSkin
        TSimpleFormSkin
        TBitmapFormSkin
```

### TCustomFormSkin

#### Methods

protected

```
procedure HitArea(X,Y: Integer;
  var Area: THitArea); virtual;
```

This method returns type of window area in Area variable. THitArea type defines system areas of window, as caption bar, caption buttons, resize borders, etc. By default, Area variable contains default area, defined by WM\_NCHITTEST event (see THitArea constants and hit test codes equivalents below). This method just calls OnHitArea event, if it is defined. You can override it in your own component.

THitArea type is defined as:

```
THitArea = (  
    haNone,  
    haCaptionBar,  
    haSysMenu,  
    haMinimizeButton,  
    haMaximizeButton,  
    haCloseButton,  
    haTopBorder,  
    haBottomBorder,  
    haLeftBorder,  
    haRightBorder,  
    haTopLeftCorner,  
    haTopRightCorner,  
    haBottomLeftCorner,  
    haBottomRightCorner,  
    haGrowBox);
```

haNone	HTNOWHERE
haCaptionBar	HTCAPTION
haSysMenu	HTSYSMENU
haMinimizeButton	HTREDUCE
haMaximizeButton	HTZOOM
haCloseButton	HTCLOSE
haTopBorder	HTTOP
haBottomBorder	HTBOTTOM
haLeftBorder	HTLEFT
haRightBorder	HTRIGHT
haTopLeftCorner	HTTOPLEFT
haTopRightCorner	HTTOPRIGHT
haBottomLeftCorner	HTBOTTOMLEFT
haBottomRightCorner	HTBOTTOMRIGHT
haGrowBox	HTGROWBOX

```
function CreateRegion: HRGN; virtual;
```

This method is called when region for parent window must be created. By default, this method calls CreateCaptionRegion, CreateBorderRegion, CreateControlRegion and CreateSkinRegion to get and merge sub-regions. You can override it to create your own transparency areas.

```
function CreateCaptionRegion: HRGN; virtual;
```

This method is called when region for window caption must be created. You can override this method to change shape of window caption.

```
function CreateBorderRegion: HRGN; virtual;
```

This method is called when region for window borders must be created. You can override this method to change shape of window borders.

```
function CreateControlsRegion: HRGN; virtual;
```

This method is called when region for window children must be created. You can override this method to change this region.

```
function CreateSkinRegion: HRGN; virtual;
```

This method is called when region for window client area must be created. By default this method calls IsTransparent method for each point of client area.

Notes.

- 1) TCustomFormSkin releases all regions, returned by all functions ahead itself.
- 2) To get more information about Window regions see “Region Reference” in WinAPI help.

```
function GetSkinWidth: Integer; virtual;
```

```
function GetSkinHeight: Integer; virtual;
```

These functions returns width and height for form skin, that placed in window client area.

```
function IsTransparent(X,Y: Integer): Boolean; virtual;
```

This method is called when TCustomFormSkin needs to know transparency of point with X and Y coordinates. By default this method calls OnTransparency event if it is defined.

```
function IsTransparentControl(  
    Control: TControl): Boolean; virtual;
```

This method is called when TCustomFormSkin needs to know transparency of the control. By default this method calls OnControlTransparency event if it is defined.

## **public**

```
destructor Destroy; override;
```

Standard destructor overridden to release system resources.

```
procedure Update; virtual;
```

This method calls internally when window region must be created. Call this method manually in your own components if you need to update region.

## Properties

protected

property Options: TSkinOptions;

This property describes options of the component. TSkinOptions is defined as

```
TSkinOption = (  
    soCaption,  
    soBorder,  
    soControls,  
    soSkin,  
    soAutoSize,  
    soClientDrag);
```

soCaption      window has visible caption

soBorder       window has visible borders

soControls     window has visible controls

soSkin          window has skin for client area

soAutoSize     window client area size is defined by skin size

soClientDrag   window can be dragged by client area

property Active: Boolean;

This property contains skin activity state.

property PopupMenu: TPopupMenu;

This property contains popup menu that is called when user clicks on the area, defined as haMenuIcon.

## Events

protected

property OnHitArea: THitAreaEvent;

This event occurs when the component needs to know area type.

```
THitAreaEvent = procedure(Sender: TObject;  
    X,Y: Integer; var Area: THitArea) of object;
```

property OnTransparency: TTransparencyEvent;

This event occurs when the component needs to know transparency of the window client area point.

```
TTransparencyEvent = procedure(Sender: TObject;  
    X,Y: Integer; var Transparent: Boolean) of object;
```

property OnControlTransparency:  
 TControlTransparencyEvent;

This event occurs when the component needs to know transparency of the control.

```
TControlTransparencyEvent = procedure(Sender: TObject;  
    Control: TControl;  
    var Transparent: Boolean) of object;
```

## **TSimpleFormSkin**

### **Properties**

#### **published**

```
property Options;  
property Active;  
property PopupMenu;
```

### **Events**

#### **published**

```
property OnHitArea;  
property OnTransparency;  
property OnControlTransparency;
```

TSimpleFormSkin just publishes some methods and events, inherited from TCustomFormSkin.

## **TBitmapFormSkin**

### **Methods**

#### **protected**

```
function GetSkinWidth: Integer; override;  
function GetSkinHeight: Integer; override;  
function IsTransparent(X,Y: Integer): Boolean; override;  
function IsTransparentControl(  
    Control: TControl): Boolean; override;
```

These internal protected methods overridden for realizing bitmap skin.

#### **public**

```
constructor Create(AOwner: TComponent); override;  
destructor Destroy; override;  
Constructor and destructor overridden for creating and destroying internal objects.
```

```
procedure Update; override;  
This method overridden for processing new properties values.
```

## Properties

### published

property TransparentColor: TColor;

This property determines which color of the Skin bitmap is to be transparent.

property Skin: TBitmap;

This property contains bitmap for form skin. All pixels with TransparentColor are transparent when Active is True.

property Preview: Boolean;

Set this property to True if you want to see Skin bitmap in design-time.

property Options;

property Active;

property PopupMenu;

Properties, inherited from TCustomFormSkin.

## Events

### published

property OnHitArea;

property OnControlTransparency;

Events, inherited from TCustomFormSkin.



## **Installing and uninstalling**

### **Registered version:**

#### Installing

1. Use "Component|Install Component..." menu item of Delphi IDE to open install dialog
2. Browse FormSkin.pas file
3. Choose any existing package or create new package
4. Click "OK" button to install the component

#### Uninstalling

1. Open package, contained FormSkin.pas
2. Delete FormSkin.pas from units list
3. Recompile package to delete component

### **Trial version**

#### Installing

- 1) Delphi
  - a) Use "File|Open..." to open GFSTRIAL.DPK
  - b) Click "Install" button in "Package" window
- 2) C++ Builder
  - a) Use "Component|Install packages..." to open "Packages" dialog box
  - b) Click "Add..." button
  - c) Locate GFSTRIAL.BPL
  - d) Click "OK" button to install package into IDE

#### Uninstalling

- 1) Use "Component|Install packages..." to open "Packages" dialog box
- 2) Choose "Gratis Form Skin Trial Package" in "Design packages" list
- 3) Click "Remove" button
- 4) Click "Yes" button to confirm removing
- 5) Click "OK" button to close dialog

## **License Agreements**

The following actions ARE ALLOWED, provided clear and prominent acknowledgement is made that "Form Skin" was used in constructing the software, and that "Form Skin" is a product of "Greatis Software":

- 1) The distribution of the products or source codes produced utilizing Form Skin.
- 2) The distribution of products or source codes produced utilizing a modified version of Form Skin.
- 3) The creation and distribution of descendant components in source code or compiled form.

The following actions ARE NOT ALLOWED:

- 1) The distribution of Form Skin, in either source or compiled form.
- 2) The inclusion of Form Skin, in source or compiled form, within other software distributions.

## Contact information

### Greatis Form Skin

Home page:	<a href="http://www.greatis.com/formskin.htm">www.greatis.com/formskin.htm</a>
Newest documentation	<a href="http://www.greatis.com/formskin.pdf">www.greatis.com/formskin.pdf</a>
Free demo	<a href="http://www.greatis.com/formskindemo.zip">www.greatis.com/formskindemo.zip</a>
Order page	<a href="http://www.greatis.com/formskinbuy.htm">www.greatis.com/formskinbuy.htm</a>
Support e-mail:	<a href="mailto:b-team@greatis.com">b-team@greatis.com</a>

### Greatis Software

Home page:	<a href="http://www.greatis.com">www.greatis.com</a>
------------	--