TGradientForm Library {button &Top/Units,JI(`',`IDH_Library_TGradientForm')} {button &Classes,JI(`',`IDH_Library_TGradientForm_Classes')} {button &Types,JI(`',`IDH_Library_TGradientForm_OtherTypes')} {button &Routines,JI(`',`IDH_Library_TGradientForm_Routines')}

Units
GradForm

Classes
TGradientForm

Other Types
TGFOnCaptionPaint
TGFPaintWhen

Routines
register

GradForm Unit {button &Top,JI(`',`IDH_Unit_GradForm')} {button &Classes,JI(`',`IDH_UnitTopic_GradForm_Classes')} {button &Types,JI(`',`IDH_UnitTopic_GradForm_OtherTypes')} {button &Routines,JI(`',`IDH_UnitTopic_GradForm_Routines')} {button &Const,JI(`',`IDH_UnitTopic_GradForm_GlobalConstants')}

Description
This unit provides the TGradientForm class, and all supporting elements.

Classes

TGradientForm
TGradientForm is a descendant of the TForm class that paints it's caption bar in a gradient fill pattern, like the Microsoft Office applications.

Other Types

TGFOnCaptionPaint
Describes the parameters used by an OnCaptionPaint event handler.

TGFPaintWhen
This enumerated type is used by the PaintGradient property to indicate when the caption should be painted as a gradient.

Routines

register
Registers the class for use in the Delphi 3 IDE.

Global Constants

DEF_GRADIENT_COLORS
The default number of colors for the GradientColors property.

MAX_GRADIENT_COLORS
The maximum number of colors that can be assigned to the GradientColors property.

MIN_GRADIENT_COLORS
The minimum number of colors that can be assigned to the GradientColors property.

TGradientForm Object {button &Top,JI(`',`IDH_Class_TGradientForm')} {button &Properties,JI(`',`IDH_ClassTopic_TGradientForm_Properties')} {button &Methods,JI(`',`IDH_ClassTopic_TGradientForm_Methods')} {button &Events,JI(`',`IDH_ClassTopic_TGradientForm_Events')}

Properties Methods Events

Unit
GradForm

Declaration
```
TGradientForm = class(TForm)
```

Description
TGradientForm is a descendant of the TForm class that paints it's caption bar in a gradient fill pattern, like the Microsoft Office applications. By default, it starts with black and moves gradually to the system defined caption color, although you can override these values. Also provided is an event to allow you to add your own custom painting on the caption bar.

Introduced Public Properties

Caption
Caption specifies a text string that appears in the caption bar.

CaptionTextColor
CaptionTextColor is the color that should be used for the text draw in the caption bar.

GradientColors
Determines the number of colors used to paint the gradient pattern.

GradientStartColor
The leftmost gradient color.

GradientStopColor
The rightmost gradient color.

PaintGradient
Determines if and when the gradient caption should be painted.

Introduced Public Methods

Create Override
Create creates and initializes an instance of TGradientForm.

Destroy Override
Destroy destroys an instance of TGradientForm.

Draw Virtual
This procedure is used to paint the caption gradient.

Introduced Public Events

OnCaptionPaint
This event is fired after the icon, buttons and gradient are painted, but just before the text is painted.

Caption property

Applies to
TGradientForm

Declaration
```
Property Caption : string;
```

Description
Caption specifies a text string that appears in the caption bar.

CaptionTextColor property

Applies to
TGradientForm

Declaration
```
Property CaptionTextColor : TColor;
```

Description
CaptionTextColor is the color that should be used for the text draw in the caption bar. You may have to adjust this color if you change the GradientStartColor to something other than the default of clBlack.

{button ,AL("GradientStartColor;GradientStopColor;Caption")} Related Topics

GradientColors property

Applies to
TGradientForm

Declaration
```
Property GradientColors : integer;
```

Description
Determines the number of colors used to paint the gradient pattern. The individual colors are determined by fading the start color into the stop color. The number of times this is done is controled by this property. The higher the number of colors, the smoother the gradient will appear. However, the more colors that are used, the more complex the painting will be.

{button ,AL("MAX_GRADIENT_COLORS;MIN_GRADIENT_COLORS;DEF_GRADIENT_COLORS")}
Related Topics

GradientStartColor property

Applies to
TGradientForm

Declaration
```
Property GradientStartColor : TColor;
```

Description
The leftmost gradient color. This is the color that is used at the beginning of the caption (the far left), and is gradually faded into the GradientStopColor.

{button ,AL("GradientStopColor;GradientColors")} Related Topics

GradientStopColor property

Applies to
TGradientForm

Declaration
```
Property GradientStopColor : TColor;
```

Description
The rightmost gradient color. This is the color that is used at the end of the caption (the far right), and is gradually faded from the GradientStartColor.

{button ,AL("GradientStartColor;GradientColors")} Related Topics

PaintGradient property

Applies to
TGradientForm

Declaration
```
Property PaintGradient : TGFPaintWhen;
```

Description
Determines if and when the gradient caption should be painted.

- gfpAlways The gradient should always be drawn.
- gfpActive Only draw the gradient when the form is active.
- gfpNever Never draw the gradient.

CalculateColors method

Applies to
TGradientForm

Declaration
```
Procedure CalculateColors;
```
Virtual

Create method

Applies to
TGradientForm

Declaration
```
Constructor Create(AOwner: TComponent);
```
Override

Description
Create creates and initializes an instance of TGradientForm. Call Create to instantiate a TGradientForm at runtime. After calling the inherited constructor, Create initializes the following properties:

- UsingDefaultGradientStopColor to TRUE.
- CaptionTextColor to clWhite.
- GradientStartColor to clBlack.
- GradientStopColor to clActiveCaption.
- GradientColors to DEF_GRADIENT_COLORS.
- PaintGradient to gpfAlways.

CreateCaptionFont method

Applies to
TGradientForm

Declaration
```
Procedure CreateCaptionFont;
```

CreateWnd method

Applies to
TGradientForm

Declaration
```
Procedure CreateWnd;
Override
```

Destroy method

Applies to
TGradientForm

Declaration
```
Destructor Destroy;
Override
```

Description
Destroy destroys an instance of TGradientForm. Do not call Destroy directly in an application. Instead, call Free. Free verifies that the instance is not already freed, and only then calls Destroy.
Destroy is used to free resources allocated in the Create constructor.

DestroyWnd method

Applies to
TGradientForm

Declaration
```
Procedure DestroyWnd;
Override
```

Draw method

Applies to
TGradientForm

Declaration
```
Procedure Draw(Active: boolean);
```
Virtual

Description
This procedure is used to paint the caption gradient. It is normally called internally, but it can be used any time a repaint of the caption is needed. The **Active** parameter is used to indicate whether the caption should be painted as the active window or an inactive window.

DrawCaption method

Applies to
TGradientForm

Declaration
```
Function DrawCaption(FormDC: HDC; Active: boolean): TRect;
```

FillRectGradientHigh method

Applies to
TGradientForm

Declaration
```
Procedure FillRectGradientHigh(DC: HDC; const R: TRect; Active: boolean);
```

FillRectGradientLow method

Applies to
TGradientForm

Declaration
```
Procedure FillRectGradientLow(DC: HDC; const R: TRect; Active: boolean);
```

FillRectSolid method

Applies to
TGradientForm

Declaration
```
Procedure FillRectSolid(DC: HDC; const R: TRect; Active: boolean);
```

GetCaptionRect method

Applies to
TGradientForm

Declaration
```
Function GetCaptionRect: TRect;
```

GradClientWndProc method

Applies to
TGradientForm

Declaration
```
Procedure GradClientWndProc(var message: TMessage);
```

IsActiveWindow method

Applies to
TGradientForm

Declaration
```
Function IsActiveWindow: boolean;
```

PaintCaptionButtons method

Applies to
TGradientForm

Declaration
```
Procedure PaintCaptionButtons(DC: HDC; var Rect: TRect);
```

PaintCaptionText method

Applies to
TGradientForm

Declaration

```
Procedure PaintCaptionText(DC: HDC; R: TRect);
```

PaintMenuIcon method

Applies to
TGradientForm

Declaration

```
Procedure PaintMenuIcon(DC: HDC; var R: TRect; Active: boolean);
```

SetCaptionText method

Applies to
TGradientForm

Declaration
```
Procedure SetCaptionText(const Val: string);
```

SetCaptionTextColor method

Applies to
TGradientForm

Declaration
```
Procedure SetCaptionTextColor(Color: TColor);
```

SetGradientColors method

Applies to
TGradientForm

Declaration
```
Procedure SetGradientColors(Val: integer);
```

SetGradientStartColor method

Applies to
TGradientForm

Declaration

```
Procedure SetGradientStartColor(Color : TColor);
```

SetGradientStopColor method

Applies to
TGradientForm

Declaration

```
Procedure SetGradientStopColor(Color : TColor);
```

SetPaintGradient method

Applies to
TGradientForm

Declaration
```
Procedure SetPaintGradient(Val: TGFPaintWhen);
```

WMEnterIdle method

Applies to
TGradientForm

Declaration
```
Procedure WMEnterIdle(var Msg: TWMEnterIdle);
Message WM_ENTERIDLE
```

WMGetText method

Applies to
TGradientForm

Declaration
```
Procedure WMGetText(var Msg: TWMGetText);
Message WM_GETTEXT
```

WMGetTextLength method

Applies to
TGradientForm

Declaration
```
Procedure WMGetTextLength(var Msg: TWMGetTextLength);
```
Message WM_GETTEXTLENGTH

WMNCActivate method

Applies to
TGradientForm

Declaration
```
Procedure WMNCActivate(var Msg: TWMNCActivate);
```
Message WM_NCACTIVATE

WMNCLButtonDown method

Applies to
TGradientForm

Declaration
```
Procedure WMNCLButtonDown(var Msg: TWMNCLButtonDown);
```
Message WM_NCLBUTTONDOWN

WMNCPaint method

Applies to
TGradientForm

Declaration
```
Procedure WMNCPaint(var Msg: TMessage);
```
Message WM_NCPAINT

WMSetCursor method

Applies to
TGradientForm

Declaration
```
Procedure WMSetCursor(var Msg: TWMSetCursor);
```
Message WM_SETCURSOR

WMSetText method

Applies to
TGradientForm

Declaration
```
Procedure WMSetText(var Msg: TWMSetText);
Message WM_SETTEXT
```

WMSettingChange method

Applies to
TGradientForm

Declaration
```
Procedure WMSettingChange(var Msg: TMessage);
```
Message WM_SETTINGCHANGE

WMSize method

Applies to
TGradientForm

Declaration
```
Procedure WMSize(var Msg: TWMSize);
Message WM_SIZE
```

WMSysColorChange method

Applies to
TGradientForm

Declaration
```
Procedure WMSysColorChange(var Msg: TWMSysColorChange);
```
Message WM_SYSCOLORCHANGE

WMSysCommand method

Applies to
TGradientForm

Declaration
```
Procedure WMSysCommand(var Msg: TWMSysCommand);
```
Message WM_SYSCOMMAND

OnCaptionPaint Event

Applies to
TGradientForm

Declaration
```
TGFOnCaptionPaint = Procedure(Sender: TObject; const Canvas: TCanvas;var R: TRect) of object;
```

Description
This event is fired after the icon, buttons and gradient are painted, but just before the text is painted. It is not fired if the caption is painted but not as a gradient, that is if PaintGradient is gfpNever or gfpActive and the form is not active.

Sender is the TGradientForm that is being painted.

Canvas is the drawing surface that is being painted. Anything you want to appear on the caption must be drawn on this canvas. This canvas is not the actual caption canvas, it is a memory bitmap (non-visible). This prevents flicker as many things are being drawn since the actual visible drawing only happens when the entire drawing operation is complete.

R is a rectangle that describes the area in which you can draw. When the event is first fired, this rectangle will be the entire caption less the system icon on the left (if any) and the caption buttons on the right (if any). After performing your drawing operations, this value should be modified so that the area you have painted is subtracted out. This prevents the gradient from painting over what you have just done.

register routine

Unit
GradForm

Declaration
```
Procedure register;
```

Description
Registers the class for use in the Delphi 3 IDE. This applies **ONLY** to Delphi 3. Previous versions of Delphi and C++Builder 1.0 do **NOT** support design-time access of TForm descendants. Sorry, just be happy Borland added it to Delphi 3 (and I presume C++B 2.0). Unlike a normal component, TForm descendant classes must also have a package installed for the registration process to work. See the installation notes in GradForm.Txt for complete instructions.

TGFOnCaptionPaint type

Unit
GradForm

Declaration
```
TGFOnCaptionPaint = procedure(Sender: TObject; const Canvas: TCanvas;
     var R: TRect) of object;
```

Description
Describes the parameters used by an OnCaptionPaint event handler.

Sender is the TGradientForm that is being painted.

Canvas is the drawing surface that is being painted. Anything you want to appear on the caption must be drawn on this canvas. This canvas is not the actual caption canvas, it is a memory bitmap (non-visible). This prevents flicker as many things are being drawn since the actual visible drawing only happens when the entire drawing operation is complete.

R is a rectangle that describes the area in which you can draw. When the event is first fired, this rectangle will be the entire caption less the system icon on the left (if any) and the caption buttons on the right (if any). After performing your drawing operations, this value should be modified so that the area you have painted is subtracted out. This prevents the gradient from painting over what you have just done.

{button ,AL("OnCaptionPaint")} Related Topics

TGFPaintWhen type

Unit
GradForm

Declaration
```
TGFPaintWhen = (gfpAlways, gfpActive, gfpNever);
```

Description
This enumerated type is used by the PaintGradient property to indicate when the caption should be painted as a gradient.

- gfpAlways The gradient should always be drawn.
- gfpActive Only draw the gradient when the form is active.
- gfpNever Never draw the gradient.

{button ,AL("PaintGradient")} Related Topics

DEF_GRADIENT_COLORS global constant

Unit
GradForm

Declaration
```
DEF_GRADIENT_COLORS = 64;
```

Description
The default number of colors for the GradientColors property. This is a good compromise between speed and appearance.

{button ,AL("MAX_GRADIENT_COLORS;MIN_GRADIENT_COLORS")} Related Topics

MAX_GRADIENT_COLORS global constant

Unit
GradForm

Declaration
```
MAX_GRADIENT_COLORS = 512;
```

Description
The maximum number of colors that can be assigned to the GradientColors property. Any more than 512 colors is not noticeable, and just slows the painting down.

{button ,AL("MIN_GRADIENT_COLORS;DEF_GRADIENT_COLORS")} Related Topics

MIN_GRADIENT_COLORS global constant

Unit
GradForm

Declaration
```
MIN_GRADIENT_COLORS = 8;
```

Description
The minimum number of colors that can be assigned to the GradientColors property. Any less than 8 colors doesn't look much like a gradient.

{button ,AL("MAX_GRADIENT_COLORS;DEF_GRADIENT_COLORS")} Related Topics

Introduced Properties
Caption
CaptionTextColor
GradientColors
GradientStartColor
GradientStopColor
PaintGradient

Introduced Methods
Create
Destroy
Draw

Introduced Events
OnCaptionPaint