



TDIBDrawingSurface Component

[Properties](#)

[Methods](#)

[Events](#)

Unit

DIBDrawingSurface

Description

The TDIBDrawingSurface component provides a visual surface for rendering Windows [DIBs](#). The current implementation of TDIBDrawingSurface uses the WING32 library from Microsoft to create and manage the DIB surface. The component creates a DIB that can be quickly blitted to the screen via the [DIBRefresh](#) method.

You can access the image data through the [DIBCanvas](#) property. You will normally refresh the surface repetitively within a fast timer event, to produce animation effects. If you want the background of the surface to be filled with a color prior to every refresh, set the [AutoBlank](#) property to True, and assign a value to [AutoBlankColor](#). If you want a background image to be rendered, set the [BackgroundDIB](#) property to an existing [TDIB](#) on the form.

The [OnCustomPaint](#) event is your hook to perform any special rendering. This event is triggered during each [DIBRefresh](#), and occurs after the background is rendered but before the image is blitted to the screen.

One or more [ISpriteEngine](#) components can be hooked to the TDIBDrawingSurface for sprite management.

Properties

[AutoBlank](#)
[BitmapInfo](#)
[DirtyRectangle](#)
[OffsetX](#)
[PhysicalWidth](#)
[WrapVertical](#)

[AutoBlankColor](#)
[ColorPalette](#)
[ForceRefresh](#)
[OffsetY](#)
[PhysicalHeight](#)

[BackgroundDIB](#)
[DIBCanvas](#)
[GDICanvas](#)
[Palette](#)
[WrapHorizontal](#)

Methods

DIBRefresh

FillSurface

RenderBackgroundDIB

Events

OnCustomPaint

Device Independent Bitmap

BitmapInfo Property

Applies To

TDIBDrawingSurface

Declaration

```
property BitmapInfo: TBitmapInfo;
```

Description

Returns the Windows TBITMAPINFO structure that describes the DIB.

GDICanvas Property

Applies To

TDIBDrawingSurface

Declaration

```
property GDICanvas: TCanvas;
```

Description

The GDICanvas property returns the standard Windows TCanvas (Device Context) that is associated with the drawing surface. You can use any standard TCanvas methods to draw on this canvas, and the results will be reflected in the final image. Generally you should avoid using these methods, as doing so will result in performance penalties.

If you do use the GDICanvas, you should first call SelectPalette and RealizePalette API functions so the correct palette will be used by the DC. Use the Palette property as the palette handle in these calls. If you wish to obtain a specified color index, you can use the API PaletteIndex call. Here's an example:

```
procedure TfrmMain.DIBDrawingSurface1CustomPaint(Sender: TObject);
begin
  with DIBDrawingSurface1 do
  begin
    SelectPalette( GDICanvas.Handle, Palette, FALSE );
    RealizePalette( GDICanvas.Handle );
    { Use GDI to draw text on the surface }
    GDICanvas.Font.Color := PaletteIndex( 100 );
    GDICanvas.TextOut( 20, 20, 'GDI Text' );
  end;
end;
```

ForceRefresh Property

Applies To

TDIBDrawingSurface

Declaration

```
property ForceRefresh: boolean;
```

Description

This property only has an effect when DirtyRectangle is True. Setting ForceRefresh to True causes the entire surface to be repainted on the next call to DIBRefresh, even if the dirty rectangle system is in use. After the next call to DIBRefresh, the property will revert back to FALSE. TDIBDrawingSurface automatically sets this to True whenever the form needs to be repainted.

Palette Property

Applies To

TDIBDrawingSurface

Declaration

```
property Palette: HPALETTE;
```

Description

The Palette property returns the logical palette handle that is being used by the underlying DIB. Even though you can link a TColorPalette to the TDIBDrawingSurface (using the ColorPalette property), it still constructs and uses its own logical palette. This is because the TColorPalette link is not required, and if it is absent the component creates a logical palette based on the current system palette.

PhysicalWidth Property

Applies To

TDIBDrawingSurface

Declaration

```
property PhysicalWidth: integer;
```

Description

The PhysicalWidth property returns the width of the underlying DIB. This could be different from the Width of the component itself. If the dimensions of the component match the dimensions of the DIB exactly, the image can be quickly blitted to the screen during DIBRefresh. Otherwise the image must be stretched, resulting in a significant performance penalty. This information is taken directly from the component's DIBCanvas property, and is provided here for convenience only.

PhysicalHeight Property

Applies To

TDIBDrawingSurface

Declaration

```
property PhysicalHeight: integer;
```

Description

The PhysicalHeight property returns the height of the underlying DIB. This could be different from the Height of the component itself. If the dimensions of the component match the dimensions of the DIB exactly, the image can be quickly blitted to the screen during DIBRefresh. Otherwise the image must be stretched, resulting in a significant performance penalty. This information is taken directly from the component's DIBCanvas property, and is provided here for convenience only.

AutoBlank Property

Applies To

TDIBDrawingSurface

Declaration

```
property AutoBlank: boolean;
```

Description

If AutoBlank is set the True, the surface will be filled with the value of AutoBlankColor during each call to DIBRefresh.

AutoBlankColor Property

Applies To

TDIBDrawingSurface

Declaration

```
property AutoBlankColor: byte;
```

Description

If AutoBlank is set to True, the surface will be filled with the value of this color index during each DIBRefresh. This byte value is an index into the 256 color palette being used by the component.

BackgroundDIB Property

Applies To

TDIBDrawingSurface

Declaration

```
property BackgroundDIB: TDIB;
```

Description

If an existing TDIB is assigned to this property, the DIB will be copied to the surface during each call to DIBRefresh. You must make sure that the dimensions of the BackgroundDIB exactly match those of the component's DIB (see PhysicalWidth and PhysicalHeight). Because the RenderBackgroundDIB method is used in this case, if the source and destination DIBs do not match, the image will not appear correct, and an access violation will occur if the source image is larger than the destination.

ColorPalette Property

Applies To

TDIBDrawingSurface

Declaration

```
property ColorPalette: TColorPalette;
```

Description

You can assign an existing TColorPalette to this property, and cause the underlying DIB to use the colors contained in this palette. If you do not assign a value to the ColorPalette property, the component will create a palette based on the existing system palette.

DIBCanvas Property

Applies To

TDIBDrawingSurface

Declaration

```
property DIBCanvas: TDIBCanvas;
```

Description

This property refers to the TDIBCanvas that is associated with the image data of this TDIBDrawingSurface. You can access the image data through this special type of canvas. You can use the drawing methods provided in TDIBCanvas to draw on the surface, or write your own custom drawing routines that access the Bits property of the DIBCanvas.

DirtyRectangle Property

Applies To

TDIBDrawingSurface

Declaration

```
property DirtyRectangle: boolean;
```

Description

If this property is set to True, the component employs a dirty rectangle system instead of regenerating the surface completely during each call to DIBRefresh. This property works in conjunction with the like property of TSpriteEngine. You should use the dirty rectangle system if your application has relatively few sprites, or few sprites that move simultaneously.

OffsetX Property

Applies To

TDIBDrawingSurface

Declaration

```
property OffsetX: integer;
```

Description

The OffsetX property allows you to manage a logical coordinate system that is large than the physical size of the drawing surface. TSpriteEngine uses these offset properties to correctly position sprites on the surface.

OffsetY Property

Applies To

TDIBDrawingSurface

Declaration

```
property OffsetY: integer;
```

Description

The OffsetY property allows you to manage a logical coordinate system that is large than the physical size of the drawing surface. TSpriteEngine uses these offset properties to correctly position sprites on the surface.

WrapHorizontal Property

Applies To

TDIBDrawingSurface

Declaration

```
property WrapHorizontal: boolean;
```

Description

If WrapHoriaontal is set to True, any sprites that move off of the left or right edges of the surface will wrap to the other side.

WrapVertical Property

Applies To

TDIBDrawingSurface

Declaration

```
property WrapVertical: boolean;
```

Description

If WrapVertical is set to True, any sprites that move off of the top or bottom edges of the surface will wrap to the other side.

DIBRefresh Method

Applies To

TDIBDrawingSurface

Declaration

```
procedure DIBRefresh;
```

Description

Calling DIBRefresh causes the DIB image to be blitted to the screen. If the dimensions of the DIB exactly matches those of the component, the blit will be very fast. Otherwise, a stretch will be performed, resulting in poorer performance.

The first task of DIBRefresh is to render the background of the surface. If AutoBlank is True, the surface is filled with the value of AutoBlankColor. If a TDIB is assigned to the BackgroundDIB property, the background image is copied to the drawing surface.

After the background is rendered, the OnCustomPaint event is triggered. It is here that you would code any custom drawing that you want to perform on the surface. This is the place to also perform any TSpriteEngine processing, such as ProcessSprites, RenderSprites, and collision detection.

After the OnCustomPaint is completed, the resulting image is copied to the screen.

Normally you will place the call to DIBRefresh within a fast timer event to produce animation effects.

FillSurface Method

Applies To

TDIBDrawingSurface

Declaration

```
procedure FillSurface( n: byte );
```

Description

The FillSurface method fills the drawing surface with the color index specified in the n parameter. This method is employed during DIBRefresh when the AutoBlank property is True.

RenderBackgroundDIB Method

Applies To

TDIBDrawingSurface

Declaration

```
procedure RenderBackgroundDIB( dib: TDIB );
```

Description

The RenderBackgroundDIB is an optimized drawing routine that dumps the image data from the specified TDIB onto the drawing surface. The dimensions of the TDIB must exactly match those of the physical drawing surface. This method is employed during DIBRefresh if the BackgroundDIB property is assigned to a valid TDIB.

OnCustomPaint Event

Applies To

TDIBDrawingSurface

Declaration

```
property OnCustomPaint: TNotifyEvent;
```

Description

The OnCustomPaint event is triggered during each call to DIBRefresh. It is provided as a hook to perform any special image processing after the default processing, but before the image is copied to the screen. Within this event, you would code any special drawing operations, and call the processing methods of attached TSpriteEngines.

