TDIBCanvas object

Properties Methods

Unit

DIBCanvas

Description

A TDIBCanvas is the drawing surface associated with a TDIB. The TDIBCanvas has a similar interface to the standard Delphi TCanvas, but because the physical image data is available to the TDIBCanvas, much of the implementation is handled by optimized routines from the GRAFIX unit.

The TDIBCanvas exposes the image data as a pointer in the Bits property.

The Width and Height properties contain the dimensions of the DIB. The Size property contains the number of bytes of image data in the DIB.

You can access the image data with the Pixels property. The PixelsClip property provides pixel access with built-in range validation.

TDIBCanvas contains several optimized drawing methods, including Rectangle, FillRect, and LineTo. Use the CopyRect method to copy a block of image data from one TDIBCanvas to another.

Properties

Bits	BrushColorIndex	ClipRect
Height	Orientation	PenColorIndex
Pixels	PixelsClip	Size
TransparentColor	Width	

Methods

CopyRect	FillRect	LineTo
MoveTo	Rectangle	

CopyRect Method

Applies To

TDIBCanvas

Declaration

```
procedure CopyRect( Dest: TRect; Canvas: TDIBCanvas; Source: TRect );
```

Description

The CopyRect method copies a rectangular area of image data from one TDIBCanvas to another. The Dest parameter specifies the destination rectangle. The Canvas property specifies the source TDIBCanvas. The Source property specifies the rectangle of the source TDIBCanvas to copy.

Although Source and Dest are both TRects, the current release of TurboSprite does not support stretching operations when using CopyRect.

FillRect Method

Applies To
TDIBCanvas

Declaration
```
procedure FillRect( rect: Trect );
```

Description
The FillRect method fills the specified rectangle with the color specified in the BrushColorIndex property.

LineTo Method

Applies To
TDIBCanvas

Declaration
```
procedure LineTo( x, y: integer );
```

Description
The LineTo method draws a line from the current internal cursor position to the location specified in the X and Y parameters. The internal cursor position is updated after the line is drawn. The line is drawn using the color specified in the PenColorIndex property.

Currently the LineTo method supports only lines with a thickness of 1.

MoveTo Method

Applies To

TDIBCanvas

Declaration

```
procedure MoveTo( x, y: integer );
```

Description

The MoveTo method positions the internal cursor to the location specified in the X and Y parameters.

Rectangle Method

Applies To

TDIBCanvas

Declaration

```
procedure Rectangle( X1, Y1, X2, Y2: integer );
```

Description

The Rectangle method draws a rectangle with an upper left corner of (X1,Y1) and a lower right corer of (X2,Y2). The rectangle is drawn using the color specified in the PenColorIndex property.

Bits Property

Applies To

TDIBCanvas

Declaration

```
property Bits: pointer
```

Description

The Bits property returns a pointer to the physical image data. You can write your own custom drawing routines to access the image data using this pointer. The example below is a simple procedure that fills the image data with a specified color:

```
procedure FillImageData( d: TDIBCanvas; const n: byte );
var
   i: integer;
   p: PByte;
begin
{ Access the image data with a byte pointer }
   p := d.Bits;
   for i := 0 to d.Size - 1 do
      begin
         p^ := n;
         Inc( p );
      end;
end;
```

While this procedure is by no means efficient, it illustrates how to manipulate DIB image data directly using the Bits property.

ClipRect Property

Applies To
TDIBCanvas

Declaration
```
property ClipRect: TRect;
```

Description
Read Only. The ClipRect property returns a rectangle that encompasses the drawing surface of the DIBCanvas' image data.

Pixels Property

Applies To

TDIBCanvas

Declaration

```
property Pixels[x, y: integer]: byte;
```

Description

The Pixels property provides direct access to the pixels of the DIBCanvas' image data. For performance reasons, this property does not perform any range checking. When using the Pixels property, proper range checking is the responsibility of the developer. The PixelsClip property performs a range check each time a pixel is accessed.

PixelsClip Property

Applies To
TDIBCanvas

Declaration
```
property PixelsClip[x, y: integer]: byte;
```

Description
The PixelsClip property, like Pixels, provides direct access to the DIBCanvas' image data. However, PixelsClip performs a range validation with each access. Any attempt to access pixel data outside of the surface's range will fail.

BrushColorIndex Property

Applies To

TDIBCanvas

Declaration

```
property BrushColorIndex: byte
```

Description

BrushColorIndex specifies the color that will be used in fill operations. Currently, FillRect is the only method that makes use of this property. The value of this property is an index into a 256 color logical palette. You can use the TColorPalette to manage a palette.

Height Property

Applies To
TDIBCanvas

Declaration
```
property Height: integer;
```

Description
The Height property returns the physical height of the image surface.

Orientation Property

Applies To
TDIBCanvas

Declaration
```
property Orientation: TDIBOrientation;
```

Description
This property determines if the physical DIB data is stored in top-down format (orTopDown) or bottom-up format (orBottomUp). All DIBs loaded from files are bottom-up DIBs. DIB surfaces created by WING or CreateDIBSection may be either format.

TDIBDrawingSurface contains a TDIBCanvas component. At design time, you may alter the Orientation property if you wish to explicitly specify a DIB orientation for the component. Of course, to allow Windows to select the most efficient orientation that the hardware supports, leave the property set to orAuto.

The custom drawing methods provided by TDIBCanvas always take DIB orientation into account. If you write your own custom drawing routines, you should write the routine to support both bottom up and top down DIB formats.

PenColorIndex Property

Applies To
TDIBCanvas

Declaration
```
property PenColorIndex: integer;
```

Description
PenColorIndex specifies the color that will be used in drawing operations. Currently, LineTo and Rectangle are the only methods that makes use of this property. The value of this property is an index into a 256 color logical palette. You can use the TColorPalette to manage a palette.

TransparentColor Property

Applies To

TDIBCanvas

Declaration

```
property TransparentColor: byte;
```

Description

This property designates a certain color index as the ‘‘transparent’’ color. The CopyRect method uses this value to perform a transparent blit of DIB data from one surface to another. The value of this property is an index into a 256 color logical palette. You can use the TColorPalette to manage a palette.

Width Property

Applies To
TDIBCanvas

Declaration
```
property Width: integer;
```

Description
The Width property returns the physical width of the image surface.

Size Property

Applies To
TDIBCanvas

Declaration
```
property Size: integer;
```

Description
The Size property returns the number of bytes contained in the image data.