

SetDIBitsToDevice

The **SetDIBitsToDevice** function sets the pixels in the specified rectangle on the device that is associated with the destination device context using color data from a device-independent bitmap (DIB).

```
int SetDIBitsToDevice(  
    HDC hdc,                // handle of device context  
    int XDest,              // x-coordinate of upper-left corner of dest. rect.  
    int YDest,              // y-coordinate of upper-left corner of dest. rect.  
    DWORD dwWidth,          // source rectangle width  
    DWORD dwHeight,         // source rectangle height  
    int XSrc,               // x-coordinate of lower-left corner of source rect.  
    int YSrc,               // y-coordinate of lower-left corner of source rect.  
    UINT uStartScan,        // first scan line in array  
    UINT cScanLines,        // number of scan lines  
    CONST VOID * lpvBits,   // address of array with DIB bits  
    CONST BITMAPINFO * lpbmi, // address of structure with bitmap info.  
    UINT fuColorUse         // RGB or palette indices  
);
```

Parameters

hdc

Identifies the device context.

XDest

Specifies the x-coordinate, in logical units, of the upper-left corner of the destination rectangle.

YDest

Specifies the y-coordinate, in logical units, of the upper-left corner of the destination rectangle.

dwWidth

Specifies the width, in logical units, of the DIB.

dwHeight

Specifies the height, in logical units, of the DIB.

XSrc

Specifies the x-coordinate, in logical units, of the lower-left corner of the DIB.

YSrc

Specifies the y-coordinate, in logical units, of the lower-left corner of the DIB.

uStartScan

Specifies the starting scan line in the DIB.

cScanLines

Specifies the number of DIB scan lines contained in the array pointed to by the *lpvBits* parameter.

lpvBits

Points to DIB color data stored as an array of bytes.

lpbmi

Points to a **BITMAPINFO** structure that contains information about the DIB.

fuColorUse

Specifies whether the **bmiColors** member of the **BITMAPINFO** structure contains explicit red, green, blue (RGB) values or indices into a palette. The *fuColorUse* parameter must be one of the following values:

Value	Meaning
DIB_PAL_COLORS	The color table consists of an array of 16-bit indices into the currently selected logical palette.
DIB_RGB_COLORS	The color table contains literal RGB values.

Return Value

If the function succeeds, the return value is the number of scan lines set.

If the function fails, the return value is zero. To get extended error information, call [GetLastError](#).

Remarks

Optimal bitmap drawing speed is obtained when the bitmap bits are indices into the system palette.

Applications can retrieve the system palette colors and indices by calling the [GetSystemPaletteEntries](#) function. After the colors and indices are retrieved, the application can create the DIB. For more information about the system palette, see [Colors](#).

The origin of a bottom-up DIB is the lower-left corner of the bitmap; the origin of a top-down DIB is the upper-left corner.

To reduce the amount of memory required to set bits from a large device-independent bitmap on a device surface, an application can band the output by repeatedly calling **SetDIBitsToDevice**, placing a different portion of the bitmap into the *lpvBits* array each time. The values of the *uStartScan* and *cScanLines* parameters identify the portion of the bitmap contained in the *lpvBits* array.