

GetDIBits

The **GetDIBits** function retrieves the bits of the specified bitmap and copies them into a buffer using the specified format.

```
int GetDIBits(  
    HDC hdc,                // handle of device context  
    HBITMAP hbm,            // handle of bitmap  
    UINT uStartScan,        // first scan line to set in destination bitmap  
    UINT cScanLines,        // number of scan lines to copy  
    LPVOID lpvBits,         // address of array for bitmap bits  
    LPBITMAPINFO lpbi,      // address of structure with bitmap data  
    UINT uUsage              // RGB or palette index  
);
```

Parameters

hdc

Identifies the device context.

hbm

Identifies the bitmap.

uStartScan

Specifies the first scan line to retrieve.

cScanLines

Specifies the number of scan lines to retrieve.

lpvBits

Points to a buffer to receive the bitmap data. If this parameter is NULL, the function passes the dimensions and format of the bitmap to the **BITMAPINFO** structure pointed to by the *lpbi* parameter.

lpbi

Points to a **BITMAPINFO** structure that specifies the desired format for the device-independent bitmap (DIB) data.

uUsage

Specifies the format of the **bmiColors** member of the **BITMAPINFO** structure. It must be one of the following values:

Value	Meaning
DIB_PAL_COLORS	The color table should consist of an array of 16-bit indices into the current logical palette.
DIB_RGB_COLORS	The color table should consist of literal red, green, blue (RGB) values.

Return Value

If the *lpvBits* parameter is non-NULL and the function succeeds, the return value is the number of scan lines copied from the bitmap.

Windows NT:

If the *lpvBits* parameter is NULL and **GetDIBits** successfully fills the **BITMAPINFO** structure, the return value is non-zero.

If the function fails, the return value is zero.

Remarks

If the requested format for the DIB matches its internal format, the RGB values for the bitmap are copied. If the requested format doesn't match the internal format, a color table is synthesized.

copied. If the requested format doesn't match the internal format, a color table is synthesized. The following table describes the color table synthesized for each format.

Value	Meaning
1_BPP	The color table consists of a black and a white entry.
4_BPP	The color table consists of a mix of colors identical to the standard VGA palette.
8_BPP	The color table consists of a general mix of 256 colors defined by GDI. (Included in these 256 colors are the 20 colors found in the default logical palette.)
24_BPP	No color table is returned.

If the *lpvBits* parameter is a valid pointer, the first six members of the **BITMAPINFOHEADER** structure must be initialized to specify the size and format of the DIB. Note that a bottom-up DIB is specified by setting the height to a positive number, while a top-down DIB is specified by setting the height to a negative number. The bitmap's color table will be appended to the **BITMAPINFO** structure.

If *lpvBits* is NULL, **GetDIBits** examines the first member of the first structure pointed to by *lpbi*. This member must specify the size, in bytes, of a **BITMAPCOREHEADER** or a **BITMAPINFOHEADER** structure. The function uses the specified size to determine how the remaining members should be initialized.

If *lpvBits* is NULL and the bit count member of **BITMAPINFO** is initialized to zero, **GetDIBits** fills in **BITMAPINFOHEADER** or **BITMAPCOREHEADER** without the color table. This technique can be used to query bitmap attributes.

The bitmap identified by the *hbm* parameter must not be selected into a device context when the application calls this function.

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