

MaskBlt

The **MaskBlt** function combines the color data for the source and destination bitmaps using the specified mask and raster operation.

BOOL MaskBlt(

```
HDC  hdcDest,          // handle of destination device context
int  nXDest,           // x-coord. of upper-left corner of destination rectangle
int  nYDest,           // y-coord. of upper-left corner of destination rectangle
int  nWidth,           // width of source and destination rectangles
int  nHeight,          // height of source and destination rectangles
HDC  hdcSrc,           // handle of source device context
int  nXSrc,            // x-coord. of upper-left corner of source rectangle
int  nYSrc,            // y-coord. of upper-left corner of source rectangle
HBITMAP hbmMask,      // handle of monochrome bit mask
int  xMask,            // horizontal pixel offset into the mask bitmap
int  yMask,            // vertical pixel offset into the mask bitmap
DWORD dwRop           // raster operation code
);
```

Parameters

hdcDest

Identifies the destination device context.

nXDest

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

nYDest

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

nWidth

Specifies the width, in logical units, of the destination rectangle and source bitmap.

nHeight

Specifies the height, in logical units, of the destination rectangle and source bitmap.

hdcSrc

Identifies the device context from which the bitmap is to be copied. It must be zero if the *dwRop* parameter specifies a raster operation that does not include a source.

nXSrc

Specifies the logical x-coordinate of the upper-left corner of the source bitmap.

nYSrc

Specifies the logical y-coordinate of the upper-left corner of the source bitmap.

hbmMask

Identifies the monochrome mask bitmap combined with the color bitmap in the source device context.

xMask

Specifies the horizontal pixel offset for the mask bitmap specified by the *hbmMask* parameter.

yMask

Specifies the vertical pixel offset for the mask bitmap specified by the *hbmMask* parameter.

dwRop

Specifies both foreground and background ternary raster operation codes that the function uses to control the combination of source and destination data. The background raster operation code is stored in the high-order byte of the high-order word of this value; the foreground raster operation code is stored in the low-order byte of the high-order word of this value; the low-order word of this value is ignored, and should be zero. The macro **MAKEROP4** creates such combinations of foreground and background raster operation codes.

For a discussion of foreground and background in the context of this function, see the following Remarks section.

For a list of common raster operation codes, see [BitBlt](#).

Return Value

If the function succeeds, the return value is TRUE.

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

Remarks

A value of 1 in the mask specified by *hbmMask* indicates that the foreground raster operation code specified by *dwRop* should be applied at that location. A value of 0 in the mask indicates that the background raster operation code specified by *dwRop* should be applied at that location.

If the raster operations require a source, the mask rectangle must cover the source rectangle. If it does not, the function will fail. If the raster operations do not require a source, the mask rectangle must cover the destination rectangle. If it does not, the function will fail.

If a rotation or shear transformation is in effect for the source device context when this function is called, an error occurs. However, other types of transformation are allowed.

If the color formats of the source, pattern, and destination bitmaps differ, this function converts the pattern or source format, or both, to match the destination format.

If the mask bitmap is not a monochrome bitmap, an error occurs.

When an enhanced metafile is being recorded, an error occurs (and the function returns FALSE) if the source device context identifies an enhanced-metafile device context.

Not all devices support the **MaskBlt** function. An application should call the [GetDeviceCaps](#) function to determine whether a device supports this function.

If no mask bitmap is supplied, this function behaves exactly like **BitBlt**, using the foreground raster operation code.

The pixel offsets in the mask bitmap map to the point (0,0) in the source device context's bitmap. This is useful in cases where a mask bitmap contains a set of masks; an application can easily apply any one of them to a mask-block transfer task by adjusting the pixel offsets and rectangle sizes sent to **MaskBlt**.