

PlgBlt

The **PlgBlt** function performs a bit-block transfer of the bits of color data from the specified rectangle in the source device context to the specified parallelogram in the destination device context. If the given bitmask handle identifies a valid monochrome bitmap, the function uses this bitmap to mask the bits of color data from the source rectangle.

```
BOOL PlgBlt(  
    HDC hdcDest,           // handle of destination device context  
    CONST POINT * lpPoint, // vertices of destination parallelogram  
    HDC hdcSrc,             // handle of source device context  
    int nXSrc,              // x-coord. of upper-left corner of source rect.  
    int nYSrc,              // y-coord. of upper-left corner of source rect.  
    int nWidth,             // width of source rectangle  
    int nHeight,           // height of source rectangle  
    HBITMAP hbmMask,       // handle of bitmask  
    int xMask,              // x-coord. of upper-left corner of bitmask rect.  
    int yMask               // y-coord. of upper-left corner of bitmask rect.  
);
```

Parameters

hdcDest

Identifies the destination device context.

lpPoint

Points to an array of three points in logical space that identify three corners of the destination parallelogram. The upper-left corner of the source rectangle is mapped to the first point in this array, the upper-right corner to the second point in this array, and the lower-left corner to the third point. The lower-right corner of the source rectangle is mapped to the implicit fourth point in the parallelogram.

hdcSrc

Identifies the source device context.

nXSrc

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

nYSrc

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

nWidth

Specifies the width, in logical units, of the source rectangle.

nHeight

Specifies the height, in logical units, of the source rectangle.

hbmMask

Identifies an optional monochrome bitmap that is used to mask the colors of the source rectangle.

xMask

Specifies the x-coordinate of the upper-left corner of the the monochrome bitmap.

yMask

Specifies the y-coordinate of the upper-left corner of the the monochrome bitmap.

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

The fourth vertex of the parallelogram (*D*) is defined by treating the first three points (*A*, *B*, and *C*) as vectors and computing $D = B + C - A$.

If the bitmask exists, a value of 1 in the mask indicates that the source pixel color should be copied to the destination. A value of 0 in the mask indicates that the destination pixel color is not to be changed.

If the mask rectangle is smaller than the source and destination rectangles, the function replicates the mask pattern.

Scaling, translation, and reflection transformations are allowed in the source device context; however, rotation and shear transformations are not.

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

The stretching mode for the destination device context is used to determine how to stretch or compress the pixels, if that is necessary.

When an enhanced metafile is being recorded, an error occurs if the source device context identifies an enhanced-metafile device context.

The destination coordinates are transformed according to the destination device context; the source coordinates are transformed according to the source device context. If the source transformation has a rotation or shear, an error is returned.

If the destination and source rectangles do not have the same color format, **PlgBlt** converts the source rectangle to match the destination rectangle.

Not all devices support the **PlgBlt** function. For more information, see the description of the RC_BITBLT raster capability in the **GetDeviceCaps** function.

If the source and destination device contexts represent incompatible devices, **PlgBlt** returns an error.