**How To Use Windows BitBlt Function in Visual Basic Application**

[View products that this article applies to.](http://support.microsoft.com/kb/147810#appliesto)

This article was previously published under Q147810

[**SUMMARY**](javascript:void(0);)

Windows GDI.EXE has a function called BitBlt that will move the source device given by the hSrcDC parameter to the destination device given by the hDestDC parameter. This article explains in detail the arguments of the Windows BitBlt function call.

[**MORE INFORMATION**](javascript:void(0);)

To use BitBlt within a Visual Basic application:   
  
Use the following Declare statement to declare the Function in 16-bit Visual Basic. (If the declaration is placed in the General Declarations section of a Form, the keyword Private must be placed at the beginning of the declaration.)

Declare Function BitBlt Lib "GDI" (ByVal hDestDC%, ByVal X%, \_

ByVal Y%, ByVal nWidth%, ByVal nHeight%, ByVal hSrcDC%, \_

ByVal XSrc%, ByVal YSrc%, ByVal dwRop&amp;) As Integer

If you are using the 32-bit version of Visual Basic use the following Declare statement. (If the declaration is placed in the General Declarations section of a Form, the keyword Private must be placed at the beginning of the declaration.)

Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, \_

ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, \_

ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, \_

ByVal ySrc As Long, ByVal dwRop As Long) As Long

The following defines each of the formal parameters used in the Declare:

Parameter Definition

-----------------------------------------------------------------------

hDestDC Specifies the device context that is to receive the

bitmap.

X,Y Specifies the logical x-coordinate and y-coordinate of

the upper-left corner of the destination rectangle.

nWidth Specifies the width (in logical units) of the destination

rectangle and the source bitmap.

nHeight Specifies the height (in logical units) of the

destination rectangle and the source bitmap.

hSrcDC Identifies the device context from which the bitmap will be

copied. It must be NULL(zero) if the dwRop&amp; parameter

specifies a raster operation that does not include a source

XSrc Specifies the logical x-coordinate and the y-coordinate of

the upper-left corner of the source bitmap.

dwRop Specifies the raster operation to be performed as defined

below.

The following Raster operations are defined using the predefined constants found in the WINDOWS.H file supplied with the Microsoft Windows Software Development Kit (SDK). The value in the parentheses () is the value to assign to the dwRop& variable.

Code/Value (hex) Description

--------------------------------------------------------------------------

BLACKNESS (42) Turn output black.

DSINVERT(550009) Inverts the destination bitmap.

MERGECOPY(C000CA) Combines the pattern and the source bitmap using the

Boolean AND operation.

MERGEPAINT(BB0226) Combines the inverted source bitmap with the

destination bitmap using the Boolean OR operator.

NOTSRCCOPY(330008) Copies the inverted source bitmap to the destination.

NOTSRCERASE(1100A6) Inverts the result of combining the destination and

source bitmap using the Boolean OR operator.

PATCOPY(F00021) Copies the pattern to the destination bitmap.

PATINVERT(5A0049) Combines the destination bitmap with the pattern using

the Boolean XOR operator.

PATPAINT(FB0A09) Combines the inverted source bitmap with the pattern

using the Boolean OR operator. Combines the result of

this operation with the destination bitmap using the

Boolean OR operator.

SRCAND(8800C6) Combines pixels of the destination and source bitmap

using the Boolean AND operator.

SRCCOPY(CC0020) Copies the source bitmap to destination bitmap.

SRCERASE(4400328) Inverts the destination bitmap and combines the

results with the source bitmap using the Boolean AND

operator.

SRCINVERT(660046) Combines pixels of the destination and source bitmap

using the Boolean XOR operator.

SRCPAINT(EE0086) Combines pixels of the destination and source bitmap

using the Boolean OR operator.

WHITENESS(FF0062) Turns all output white.

**Step-by-Step Example**

Here is an example showing how to copy the contents of a picture control to the contents of another picture control.

1. Start a new project in Visual Basic. Form1 is created by default. Place two picture controls (Picture1 and Picture2) on Form1.
2. Add a BAS module to the project.
3. In the General Declarations section of the module place the following code to declare the BitBlt API:

Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x \_ As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As \_ Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As \_ Long, ByVal dwRop As Long) As Long

1. Display some graphics on Picture1 by loading from a picture file or by pasting from the clipboard at design time. You can load a picture from a file as follows:
   * + Select Picture from the Properties list box and click the button with three dots to the right of the Settings box.
     + Then select the desired picture file such as a .BMP or .ICO file supplied with Microsoft Windows from the dialog box.
2. Add the following code to the Form\_Click procedure:

Private Sub Form\_Click ()

Const PIXEL = 3

Picture1.ScaleMode = PIXEL

Picture2.ScaleMode = PIXEL

hDestDC& = Picture2.hDC

x& = 0: y& = 0

nWidth& = Picture2.ScaleWidth

nHeight& = Picture2.ScaleHeight

' Assign information of the source bitmap.

hSrcDC& = Picture1.hDC

xSrc& = 0: ySrc& = 0

' Assign the SRCCOPY constant to the Raster operation.

dwRop& = &HCC0020

Suc& = BitBlt(hDestDC&, x&, y&, nWidth&, nHeight&,

hSrcDC&, \_ xSrc&, ySrc&, dwRop&)

End Sub

1. Run the program. Click the form. The contents of the first picture will be displayed on the se cond picture.