# Load Bitmap to Desired Size

# Table of Contents

## LOADBMP

Liberty BASIC's LOADBMP command allows us to load bitmaps into memory. We can then draw them in graphics windows or graphicboxes with the DRAWBMP command. They are displayed in the size in which they were saved to disk. If a bitmap is 100x100 pixels, then it is displayed at that size. LOADBMP looks like this when a diskfile is loaded:

```
loadbmp "bmpname", filename.bmp
```

You can also load images with API calls using either add-on DLLs or the Windows API. If you load an image with an API call, it returns the handle of the loaded image. Use that handle with LOADBMP like this:

```
'bitmap is loaded by API as hBitmap
loadbmp "bmpname", hBitmap
```

## LoadImageA

The LoadImageA API call can load an image on disk into memory. The first argument of the function is for the instance handle. This is not used when loading an image from disk, so we pass it as 0.

The second argument is a string containing the disk filename of the image.

The next argument specifies the type of image. For bitmaps, the Windows constant "_IMAGE_BITMAP" is used.

The next two arguments are for the desired width and height of the image. If these arguments are passed as 0, the image is loaded at the same dimensions as it was saved. If these arguments contain values, the image is loaded at those dimensions.

The final argument is the load flag. Since we are loading a file from disk, we use the Windows constant, "_LR_LOADFROMFILE".

The function returns the memory handle of the loaded bitmap if it is successful, 0 if it fails.

```
calldll #user32, "LoadImageA",_
0 as ulong,_                   'instance - use 0 for image from file
imagePath$ as ptr,_            'path and filename of image
_IMAGE_BITMAP as long,_        'type of image
width as long,_                'desired width
height as long,_               'desired height
_LR_LOADFROMFILE as long,_     'load flag
hImage as Ulong                'handle of loaded image
```

## Loading at Desired Size

We can load an image so that it is any size we want. This does not change the disk file of the image. It only changes the size of the image in memory. We can use this method to cause a bitmap to fill a graphicbox, no matter the size of the graphicbox. The image may be distorted, since the actual width to height proportion of the image as saved to disk may differ from the one we use to load it. For instance, if an image is 100 pixels wide and 200 pixels high on disk, but we load it to be 100x100, it will appear squashed when displayed. Keep this in mind when using this method. It's a great way to create thumbnail images.

## DEMO

Here is a demonstration program. Change the width and height values at the top of the code to see how it changes the appearance of the images.

```
'Use LoadImageA to load bitmaps to be a specified size
nomainwin

width=200   'desired width
height=200  'desired height
```

```
menu #1, "&File", "&Open",[open],"E&xit",[quit]
graphicbox #1.g, 0,0,width,height
open "Image Size Test" for window as #1
#1 "trapclose [quit]"
wait

[open]
filedialog "Open Image","*.bmp", imagePath$
if imagePath$="" then wait

if hImage then unloadbmp "image"

calldll #user32, "LoadImageA",_
0 as ulong,_                 'instance - use 0 for image from file
imagePath$ as ptr,_          'path and filename of image
_IMAGE_BITMAP as long,_      'type of image
width as long,_              'desired width
height as long,_             'desired height
_LR_LOADFROMFILE as long,_   'load flag
hImage as Ulong              'handle of loaded image

if hImage = 0 then
    notice "Cannot load image."
    wait
end if

loadbmp "image", hImage      'get the LB loadbmp name

print #1.g, "down"
print #1.g, "drawbmp image 0 0"

wait

[quit]
if hImage then unloadbmp "image"
close #1:end
```