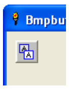
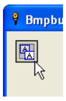
Why Create BmpButtons?

Liberty BASIC has easy, native BmpButtons. The alternate method described here is useful because the appearance of Liberty BASIC BmpButtons does not suit everybody.

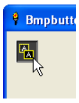
BmpButtons look like this when they are not being clicked:



Windows BmpButtons look like they are pushed down when the user clicks on them:



Liberty BASIC BmpButtons don't conform to 32-bit Windows norms in their appearance. Liberty BASIC BmpButtons invert their colors as a signal that they are depressed. (When I'm depressed, I reach for a good book!)



Stylebits to the Rescue!

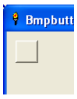
The STYLEBITS command, which is new in Liberty BASIC 4, allows us to alter the properties of a control or window. The first argument is for adding stylebits. We can alter a regular BUTTON by adding the _BS_BITMAP style to it, like this:

```
stylebits #1.b, _BS_BITMAP,0,0,0
button #1.b, "",[click],UL,10,10,25,25
```

The button doesn't need a caption, since the text won't be displayed anyway. The width and height arguments should match the width and height of the image we plan to use on our bmpbutton. If we don't include width and height arguments, Liberty BASIC sizes the button to fit the caption.

Sending a Message

The regular button has now become a BmpButton. If we stop here, it won't look very good!



We must tell the button which image to display. We do that by using the SendMessageA API call. The function requires the handle to the button. We retrieve that handle with the HWND command:

```
    hButton=hwnd(#1.b)
```

The next argument is the message to send, which is _BM_SETIMAGE. This is the message that tells the button to use the image whose handle is passed into the function. The next argument tells the function the type of image being used. We are using a bitmap, so this is _IMAGE_BITMAP. The last argument is the handle of the bitmap. We must first load a bmp with LOADBMP, then retrieve its handle with HBMP.

```
loadbmp "test","bmp\copy.bmp"
hBitmap = hbmp("test")
```

Here is the SendMessageA function as it appears in the demo program:

```
    CallDLL #user32, "SendMessageA",_
    hButton As uLong, _      'handle of button
    _BM_SETIMAGE As Long,_   'message to set new image
    _IMAGE_BITMAP as long,_  'type of image
    hBitmap As uLong,_       'handle of bitmap
    re As Long
```

DEMO

```
'run from root LB directory so program can find bmp

loadbmp "test","bmp\copy.bmp"
hBitmap = hbmp("test")

nomainwin
stylebits #1.b, _BS_BITMAP,0,0,0
button #1.b, "",[click],UL,10,10,35,35
open "Bmpbutton Play" for window as #1
#1 "trapclose [quit]"

    hButton=hwnd(#1.b)

    'set new bmp
    CallDLL #user32, "SendMessageA",_
    hButton As uLong, _       'handle of button
    _BM_SETIMAGE As Long,_    'message to set new image
    _IMAGE_BITMAP as long,_  'type of image
    hBitmap As uLong,_       'handle of bitmap
    re As Long

wait

[quit] unloadbmp "test":close #1:end

[click]
   'notice "Hello"
    wait
```