CLIPBOARD API DEMOS

- Alyce *and* - DennisMcK

Table of Contents

Clipboard API Functions

- Alyce

These three demos show you how to handle text, bitmaps and wavs with the clipboard.

Text
Bitmap
Wav

OpenClipboard

To access the Windows Clipboard via API, you must first make a call to OpenClipboard. The argument required is your windows's handle and the return is nonzero if the function successds.

```
CallDll #user32, "OpenClipboard", hWnd as ulong, r as long
```

CloseClipboard

When you have finished, you must call CloseClipboard. It will return nonzero if successful.

```
CallDll #user32, "CloseClipboard", r as long
```

SetClipboardData

To place something on the clipboard once it is open, use SetClipboardData. You must specify the format of the data, so the clipboard knows if it is receiving text or an image, for instance. Some possible formats we can use:

_CF_TEXT
_CF_BITMAP
_CF_WAVE

You must supply the memory handle to the data. If you are sending text to the clipboard, you will use memory allocation functions to create a handle to the text in memory. See the demo by Dennis below for a step-by-step example. You will use the functions "GlobalAlloc", "GlobalLock", "GlobalUnlock", and "GlobalFree".

If you are sending a bitmap to the clipboard, you will need the handle to the bitmap, retrieved with Liberty BASIC's HBMP() function. If the function succeeds, the return value is the handle of the data.

```
calldll #user32, "SetClipboardData",
    format as long,_
    handle as ulong,_
    rethandle as long
```

GetClipboardData

To retrieve data from the clipboard, use GetClipboardData. It requires the type of format you seek as the first argument. You must specify whether you are looking for text, a bitmap or a wave. If data in the specified format is on the clipboard, the function returns a handle to the data. If the return is null, that means that data in the format you specified is not available on the clipboard.

```
calldll #user32, "GetClipboardData",_
    format as long,_
    rethandle as ulong

calldll #user32, "GetClipboardData",_
    _CF_TEXT as long,_
    txtHandle as ulong

calldll #user32, "GetClipboardData",
    _CF_BITMAP as long,_
    hBmp as ulong

calldll #user32, "GetClipboardData",_
    _CF_WAVE as long,_
    hWav as ulong
```

You will need to manage the data differently depending upon the format. If you retrieve text from the clipboard, the return is a memory pointer to the text. Use the winstring() function to retrieve the actual text.

If you retrieve a bitmap from the clipboard, use the LOADBMP function to load it from the handle returned by the function. Once you have given it an LB name with LOADBMP, you can use DRAWBMP or BMPSAVE just as you would on a bitmap loaded from disk.

If you retrieve the handle of a wav from the clipboard, you cannot play it with PLAYWAVE. Since it is a wav in memory, you must play it with the API function sndPlaySound. See the third program below for examples of retrieving and using data in text, bitmap and wav format.

Placing Text on the Clipboard

- DennisMcK

```
'Send text to clipboard with api calls
'by Dennis McKinney
'Note: You can use your window handle here instead of 0
r = ClipboardSetText(0, "Hello World")
'If r is > 0 then the text has been placed on the clipboard
end
```

```
Function ClipboardSetText(hOwner, strText$)
'Puts some text on the Windows clipboard
'In: The text to place on the clipboard
'Out: 0 if fail, > 0 if text placed on clipboard

strText$ = strText$ + chr$(0)
lngSize = Len(strText$)
hMemory = GlobalAlloc(_GMEM_MOVEABLE, lngSize)

'Lock the object into memory
lpMemory = GlobalLock(hMemory)

'Move the string into the memory we locked
CallDll #kernel32,"RtlMoveMemory", lpMemory as ulong,
 strText$ as ptr, lngSize as long, ret as void

'Don't send clipboard locked memory.
r = GlobalUnlock(hMemory)

'Open the clipboard
CallDll #user32, "OpenClipboard", hOwner as ulong, r as long

'Remove the current contents of the clipboard
CallDll #user32, "EmptyClipboard", r as long

'Add our string to the clipboard as text
'If hSuccess > 0 then we have set the clipboard data
CallDll #user32, "SetClipboardData", _CF_TEXT as long,
 hMemory as ulong, hSuccess as long

'Close the clipboard
CallDll #user32, "CloseClipboard", r as long

'If we have set the clipboard data, we no longer own
'the memory--Windows does, so don't free it unless we failed.
If hSuccess = 0 Then
r = GlobalFree(hMemory)
End If

ClipboardSetText = hSuccess
End Function

'--- Supporting functions ---

Function GlobalAlloc(type, dwBytes)
CallDll #kernel32, "GlobalAlloc", type as long, dwBytes as ulong,
 GlobalAlloc as long
End Function

Function GlobalLock(hMem)
CallDll #kernel32, "GlobalLock", hMem as ulong, GlobalLock as long
End Function

Function GlobalUnlock(hMem)
CallDll #kernel32, "GlobalUnlock", hMem as ulong, GlobalUnlock as long
End Function

Function GlobalFree(hMem)
CallDll #kernel32, "GlobalFree", hMem as ulong, GlobalFree as long
End Function
```

Editor's note: an example that retrieves text from the clipboard is included in the third demo, below.

PLACE BITMAP ON CLIPBOARD AND RETRIEVE BITMAP FROM CLIPBOARD

- Alyce

```
'This little demo shows how to send a bitmap to the clipboard.
'It also shows how to grab a bitmap from the clipboard and
'display it in our Liberty BASIC program.
nomainwin
WindowWidth=340:WindowHeight=480
graphicbox #1.g, 10,10,300,300
statictext #1.s, "",10,320,600,50
open "Clipboard Demo" for window as #1
print #1, "trapclose [quit]"
h=hwnd(#1)

filedialog "Open","*.bmp",bmpfile$
if bmpfile$="" then [quit]

loadbmp "forclip",bmpfile$
hBitmap=hbmp("forclip")'get bmp handle

'open clipboard:
calldll #user32, "OpenClipboard",h as ulong, result as long
```

```
'put bmp data on clipboard:
calldll #user32, "SetClipboardData",_CF_BITMAP as long,_
    hBitmap as ulong, rethandle as ulong

'see if bmp data is on clipboard
calldll #user32, "GetClipboardData",_CF_BITMAP as long,_
    hBmp as ulong

'if bmpdata is on clipboard, load the bmp and draw it
if hBmp<>0 then
    loadbmp "demo",hBmp
    print #1.g, "down;fill lightgray;drawbmp demo 0 0;flush"
end if

calldll #user32, "CloseClipboard", result as void

msg$="Return for CF_BITMAP ";hBmp
print #1.s, msg$

wait

[quit]
if hBmp<>0 then unloadbmp "demo"
close #1:end
```

RETRIEVE TEXT, BITMAP AND WAV DATA FROM THE CLIPBOARD

- Alyce

```
'to use this demo, do one of the following:
'
'open notepad, type some text, and choose COPY
'           OR
'open MS Paint, open a picture, then select all
'or part of it and choose COPY
'           OR
'open a wav in sound recorder, then choose COPY
'
'
'if  you have copied text to the clipboard, it will
'appear in the texteditor.
```

```
'
'if you have copied an image to the clipboard,
'it will be drawn in the graphicbox
'
'if you have copied a wav, it will play
'

nomainwin
WindowWidth=640:WindowHeight=480
texteditor #1.t, 10,10,300,300
graphicbox #1.g, 320,10,300,300
statictext #1.s, "",10,320,600,50
button #1.b, "Do another!",[again],UL,10,380
open "Clipboard Demo" for window as #1
print #1, "trapclose [quit]"
h=hwnd(#1)

[again]

calldll #user32, "OpenClipboard",h as ulong, result as long
calldll #user32, "GetClipboardData",_CF_TEXT as long,_
    txt as long

if txt<>0 then
    print #1.t, "!cls"
    print #1.t, "Text content of clipboard:"
    print #1.t, ""
    print #1.t, winstring(txt)
    print #1.t, "!origin 1 1"
end if

calldll #user32, "GetClipboardData",_CF_BITMAP as long,_
    hBmp as ulong

if hBmp<>0 then
    loadbmp "demo",hBmp
    print #1.g, "cls;down;drawbmp demo 0 0;flush"
end if

calldll #user32, "GetClipboardData",_CF_WAVE as long,_
    hWav as ulong
if hWav<>0 then
    SND.SYNC = 0
    SND.ASYNC = 1
    SND.NODEFAULT = 2
    SND.MEMORY = 4
    mode=SND.MEMORY or SND.ASYNC OR SND.NODEFAULT
    calldll #winmm, "sndPlaySoundA",_
    hWav as ulong,mode as long,result as long
end if

calldll #user32, "CloseClipboard", result as void

msg$="Return for CF_TEXT ";txt
msg$=msg$+ chr$(13)+"Return for CF_BITMAP ";hBmp
msg$=msg$+chr$(13)+"Return for  CF_WAVE ";hWav
print #1.s, msg$

wait

[quit]
if hBmp<>0 then unloadbmp "demo"
close #1:end
```