

The ABCs of APIs Lesson 4

Sending Text Strings into API calls.

Table of Contents

[Strings](#)

[Strings in CALLDLL](#)

[Handles of Text and Graphics Windows](#)

[What's Next?](#)

In [Lesson 2](#) we learned how to relocate and resize a window or control with API calls. Liberty BASIC allows us to relocate and resize controls natively, but it does not have the same abilities for windows. The same is true of the caption or text string. Most controls recognize a command that changes their caption, but a window's titlebar text cannot be changed with native Liberty BASIC commands.

We can use the **SetWindowTextA** API call to send a new caption to a window as a string of text.

Strings

Strings of text can be literal strings or string variables. A string literal is contained inside of quotation marks. Here is a string literal:

"String of Text"

A string variable name ends in a dollar sign character. It is empty until you assign a string of text to it. Here are some acceptable string variable names:

Name\$

Text\$

Assign values to variables in Liberty BASIC with the 'equals' character. The two variables are assigned text like this:

```
Name$ = "John Smith"
Text$ = "Hello, World!"
```

Strings in CALLDLL

String arguments are passed as type **PTR** in a CALLDLL statement.

Type PTR is similar to struct in the way it is used, but it does not contain a list of members like a struct, but rather it is a string of characters. When a string is passed into an API function "AS PTR", a pointer (PTR) to the 32-bit memory address of the string is passed, not the string itself.

Here is a function that changes the caption on the titlebar of a window. Arguments include the handle of the window whose caption is to be modified, and the new text string. This literal string of text is passed as type PTR.

```
CallDll #user32, "SetWindowTextA", _  
    h as ulong, _           'handle of window or control  
    "New Caption" as ptr, _ 'new text string  
    result as void          'no return
```

Here is the function used to change the caption of a window in a Liberty BASIC program.

```
nomainwin  
Open "Old Caption" for window as #1  
    #1 "trapclose [quit]"  
  
    h = hwnd(#1)  
  
    CallDll #user32, "SetWindowTextA", _  
        h as ulong, _           'handle of window or control  
        "New Caption" as ptr, _ 'new text string  
        result as void          'no return  
  
wait  
  
[quit] close #1:end
```

The new caption may also be passed as a string variable:

```
NewText$ = "This text was changed!"  
calldll #user32, "SetWindowTextA", _
```

```
h as ulong, _      'handle of window or control
NewText$ as ptr, _ 'new text string
result as void     'no return
```

Here is the code in a demonstration program:

```
nomainwin
Open "Old Caption" for window as #1
  #1 "trapclose [quit]"

  h = hwnd(#1)

  NewText$ = "New Caption"
  CallDll #user32, "SetWindowTextA", _
    h as ulong, _      'handle of window or control
    NewText$ as ptr, _ 'new text string
    result as void     'no return

wait

[quit] close #1:end
```

In [Lesson 1](#) we discussed the fact that one cannot use expressions or arrays as arguments in CALLDLL. The following code will cause an error in the program because we've attempted to pass a string expression as an argument.

```
nomainwin
Open "Old Caption" for window as #1
  #1 "trapclose [quit]"

  h = hwnd(#1)

  NewText$ = "New Caption"
  CallDll #user32, "SetWindowTextA", _
    h as ulong, _
    NewText$ + "Wow!" as ptr, _
    result as void

wait

[quit] close #1:end
```

Handles of Text and Graphics Windows

In [Lesson 1](#) we learned that handles are passed as type "word" or type "ulong". We also learned that handles of Liberty BASIC windows and controls are obtained with the **HWND()** function.

We must now note that windows opened "for graphics" or "for text" are different than other Liberty BASIC windows. Retrieving the handle of these two types of windows actually gives us the handle of the workspace of the window. That is the graphics part of the graphics window and the texteditor part of the text window. To get the handle of the entire window, we must obtain the handle of the window as usual with **HWND()** then use the **GetParent** API call to retrieve the handle of the entire window. We'll use that handle to change the titlebar of a graphics window in the following demonstration code.

```
nomainwin
Open "Old Caption" for graphics as #1
    #1 "trapclose [quit]"

    h = hwnd(#1)

    callDll #user32, "GetParent", _
        h as ulong, _      'handle of graphics window
        hParent as ulong   'handle of entire window

    NewText$ = "New Caption"
    CallDll #user32, "SetWindowTextA", _
        hParent as ulong, _      'handle of parent of graphics window
        NewText$ as ptr, _      'new text string
        result as void          'no return

wait

[quit] close #1:end
```

What's Next?

[Lesson 5](#) will discuss the retrieval of text strings in API calls.