The ABCs of APIs Lesson 3

Using numeric types in API calls.

Table of Contents

Numeric Types in Liberty BASIC

BITS

Types in API Functions

What's Next?

In Lesson 2 we used the MoveWindow API call to change the size and location of a window. Have a look at the "as type" statements below.

```
    calldll #user32, "MoveWindow",_
        hWindow as ulong,_  'handle of window
        10 as long,_        'x location of window
        10 as long,_        'y location of window
        550 as long,_       'width of window
        350 as long,_       'height of window
        1 as long,_         'repaint flag, 1=yes,0=no
        result as long      'nonzero = success
```

hWindow is passed as type **ulong** while the x location of the window is passed as type **long**. Why is there a difference, and why is it important?

Numeric Types in Liberty BASIC

The helpfile shows us that Liberty BASIC uses the following numeric types in CALLDLL and STRUCTs.

double	(8 bytes, a double floating point)
ulong	(4 bytes, unsigned long integer)
long	(4 bytes, signed long integer)
short	(2 bytes, signed short integer)
ushort, word	(2 bytes, unsigned short integer)

boolean	(2 bytes, true/false expression, deprecated, translate BOOL into Long)
char[n]	(n bytes, available in structures only)

BITS

An understanding of numeric types begins with an understanding of the basic unit of information understood by a computer, the **bit**.

"Bit" is short for **BI**nary digi**T**. Binary digits are a basic unit of information storage and communication in digital computing. A bit has a value of either 0 or 1. For example, the number 10010111 is 8 bits long.

A **byte** is a unit of information storage that contains eight bits. A byte can represent 256 integer values, in the range of 0-255. In Liberty BASIC, bytes can be used in structs only and are represented by **char[n]** where 'n' is the number of bytes.

A **word** or **ushort** consists of two bytes, which is 16 bits. It can represent 65536 integer values in the range 0-65535.

In Liberty BASIC, a **boolean** type also consists of 2 bytes, but it represents a true/false expression. A value of 0 is false and any other value is true.

A **short** also consists of two bytes. It can represent 65536 integer values in the range of –32,768 to 32,767.

A **long** consists of four bytes, or 32 bits. It can contain integer values in the range of –2,147,483,648 to 2,147,483,647.

A **ulong** also consists of four bytes. It can contain integer values in the range of 0 to 4,294,967,295.

A **double** consists of eight 8 bytes. It can contain **decimal** values in the range of 1E-307 to 1E+308 (15 decimal digits).

Did you notice that some types can hold negative numbers, while other types cannot? For instance, a **short** can hold -32,768 to 32,767, while a **ushort** which is also two bytes in size can hold 0 to 65535. This happens because each of the bits in these 16 bit types can hold one piece of information. In the case of a **short** type, one of the bits is reserved for the sign. The sign can either be positive or negative. Since that bit is reserved for the sign, there are only 15 bits left to determine the size of the number. In type **ushort**, all 16 bits can be used to determine the size of the number, so it can hold a larger number. That number must be positive, though. The **u** in types **ushort** and **ulong** stands for **unsigned**.

A **double** type is the only Liberty BASIC type that can contain decimal values. **Double** is short for **double**

floating point number.

Types in API Functions

We send numeric arguments into API functions "as type" because they need to know the type information to know how to interpret the number. Failure to use the correct type in an API call can cause a function to work incorrectly or even to fail.

Look again at the code from Lesson 2 .

```
    calldll #user32, "MoveWindow",_
        hWindow as ulong,_  'handle of window
        10 as long,_        'x location of window
        10 as long,_        'y location of window
        550 as long,_       'width of window
        350 as long,_       'height of window
        1 as long,_         'repaint flag, 1=yes,0=no
        result as long      'nonzero = success
```

The hWindow argument is the handle of a window or control. Windows handles are unsigned 32 bit integers, so we pass that argument as type **ulong**.

The x and y locations of the window or control to be moved can be negative numbers, so we need to pass them as signed 32 bit integers, or type **long**.

Try it yourself! Copy the code from Lesson 2 into Liberty BASIC and try using negative numbers for the x and y locations.

What's Next?

Lesson 4 will discuss the use of text strings in API calls.