

Table of Contents

How to assign an unsigned integer value in Visual Basic 6	1
Using a Double variable	2
Using Byte variables	2

[BioStar 1](#), [Visual Basic 6](#), [Integer](#)

How to assign an unsigned integer value in Visual Basic 6

This document explains how to assign an unsigned integer value to a signed integer variable in Visual Basic 6.0. There is no unsigned integer type in Visual Basic 6.0. You have a 4-byte signed Long type and a 4-byte Double type. This can cause a problem since BioStar SDK is originally written in C++, and in Visual Studio 6.0, you need to define matching Visual Basic data structures in order to use the data structures and functions defined in the BioStar SDK DLL. There is no automatic conversion.

You need to define them on your own. However, you might have a hard time when you declare a variable that is declared as an unsigned integer in C++ in Visual Basic 6.0. As already mentioned, there is no equivalent data type for the unsigned integer type.

Take the following data structure as an example:

```
typedef struct {
    int version;
    unsigned userID;
    time_t startTime;
    time_t expiryTime;
    unsigned cardID;
    unsigned char cardCustomID;
    unsigned char commandCardFlag;
    unsigned char cardFlag;
    unsigned char cardVersion;
    unsigned short adminLevel;
    unsigned short securityLevel;
    unsigned accessGroupMask;
    unsigned short numOfFinger; // 0, 1, 2
    unsigned short fingerChecksum[2];
    unsigned char isDuress[2];
    int disabled;
    int opMode;
    int dualMode;
    char password[16]; // for BioLite Net only
    unsigned fullCardCustomID;
    int reserved2[14];
} BEUserHdr;
```

The most difficult member to declare in Visual Basic 6.0 is **cardID**. If you declare it as a **Long** variable, the ID ranges that the two variables can cover are different. A Long type variable can contain values from -2,147,483,648 to 2,147,483,647 while an **Unsigned** type variable can contain values from 0 to 4,294,967,295. Most smart cards have card numbers of 32 bits, so it's not uncommon to see card numbers greater than 2,147,483,647.

Then, how can we define **cardID** in Visual Basic 6.0? There are two possible solutions to this: using a Double variable and four byte variables. You can choose either of them.

Using a Double variable

In order to use a Double variable for unsigned integer value, you need to include conversion functions as follows:

```
Private Const OFFSET_4 = 4294967296#  
Private Const MAXINT_4 = 2147483647  
Function LongToUnsigned(ByVal Value As Long) As Double  
    If Value < 0 Then  
        LongToUnsigned = Value + OFFSET_4  
    Else  
        LongToUnsigned = Value  
    End If  
End Function  
  
Function UnsignedToLong(ByVal Value As Double) As Long  
    If Value < 0 Or Value >= OFFSET_4 Then Error 6  
    If Value <= MAXINT_4 Then  
        UnsignedToLong = Value  
    Else  
        UnsignedToLong = Value - OFFSET_4  
    End If  
End Function
```

The following code shows how to use the conversion functions in order to use 3,760,382,976 as a card ID:

```
Private Sub CommandTest_Click()  
    Dim cardID_of_UserHdr As Long  
    Dim cardID_of_VB As Double  
    'To transfer card ID to a device, convert 3760382976 to -534584320 using  
    UnsignedToLong()  
    cardID_of_UserHdr = UnsignedToLong(3760382976.0#)  
    'To use card ID transferred from a device in VB, convert -534,584,320 to  
    3,760,382,976.  
    cardID_of_VB = LongToUnsigned(cardID_of_UserHdr)  
End Sub
```

Using Byte variables

You can use a card ID greater than 2,147,483,647 by declaring four Byte variables as follows:

```
Public Type BEUserHdr  
    version As Long  
    userID1 As Byte  
    userID2 As Byte
```

```
userID3 As Byte
userID4 As Byte
startTime As Long
expiryTime As Long
cardID As Long
cardCustomID As Byte
commandCardFlag As Byte
cardFlag As Byte
cardVersion As Byte
adminLevel As Integer
securityLevel As Integer
accessGroupMask As Long
numOfFinger As Integer ' 0, 1, 2
fingerChecksum(1) As Integer
isDuress(1) As Byte
disabled As Long
opMode As Long
dualMode As Long
password(15) As Byte
reserved2(14) As Long
End Type
```

Let's say that we want to use 3,965,196,378 to a card ID. If we represent the value in a binary number, it will be 11101100 01011000 00011000 01011010. You can assign the values to the Byte variables as follows:

```
userHdr.userID1 = 90 // 01011010
userHdr.userID2 = 24 // 00011000
userHdr.userID3 = 88 // 01011000
userHdr.userID4 = 236 // 11101100
```

From:

<http://kb.supremainc.com/knowledge/> -

Permanent link:

http://kb.supremainc.com/knowledge/doku.php?id=en:1xfaq_how_to_assign_an_unsigned_integer_value_in_visual_basic_6

Last update: **2015/10/12 13:59**