

## Creating a Visual Basic 6 Module From a DLL Header File

- Create a Visual Basic module
- Declare library functions in this module using public declare ... lib for each exported function

Example:

### C (.h) File:

```
__declspec(dllexport) int Add(int a, int b);
```

### VB6 (.bas) File:

```
Public Declare Function Add Lib "Math.dll" (ByVal a As Long, ByVal b As Long) As Long
```

- Create public const variables for each #define constant

Example:

### C(.h) File:

```
#define PI 3.14159
```

```
#define MAX_BYTE_VALUE 0xFF
```

### VB6 (.bas) File:

```
Public Const PI = 3.14159
```

```
Public Const MAX_BYTE_VALUE = &HFF
```

- Use the appropriate data types:

Type	C	VB6
1 byte unsigned	bool, unsigned char, BYTE	Byte
1 byte signed	char	Byte
2 bytes unsigned	unsigned short, WORD	Integer
2 bytes signed	short	Integer
4 bytes unsigned	unsigned int, unsigned long, UINT, DWORD	Long
4 bytes signed	int, long, BOOL	Long
4 bytes floating point	float	Single
8 bytes floating point	double	Double
4/8 byte pointer	void*	Any (ByVal)

See [http://msdn.microsoft.com/en-us/library/aa263420\(VS.60\).aspx](http://msdn.microsoft.com/en-us/library/aa263420(VS.60).aspx) for more information on data types.

- Special cases:

1. Visual Basic 6 differentiates between functions that return a value and functions that return void. A function that returns a value is called a function in VB, and a function that returns void (no return value) is called a subroutine.

Example:

**C (.h) File:**

```
__declspec(dllexport) void SetLatch(int value);
```

**VB6 (.bas) File:**

```
Public Declare Sub SetLatch Lib "Some.dll" (ByVal value As Long)
```

2. Parameters passed as a pointer should use the `ByRef` keyword.

Example:

**C (.h) File:**

```
__declspec(dllexport) void Halve(BYTE* value);
```

**VB6 (.bas) File:**

```
Public Declare Sub Halve Lib "Math.dll" (ByRef value As Byte)
```

3. Parameters passed as a C string (`char*`) should use the `String` data type (by value).

Example:

**C (.h) File:**

```
__declspec(dllexport) void GetName(char* name, int size);
```

**VB6 (.bas) File:**

```
Public Declare Sub GetName Lib "Some.dll" (ByVal name As String, ByVal size As Long)
```

Calling Example:

```
Dim name as String * 128  
Call GetName(name, 128)
```

4. Parameters passed as an array should use a reference to the first element of the array.

Example:

**C (.h) File:**

```
__declspec(dllexport) void GetBuffer(BYTE* buffer, int size, int* bytesReturned);
```

**VB6 (.bas) File:**

```
Public Declare Sub GetBuffer Lib "Some.dll" (ByRef buffer As Byte, ByVal size As Long, ByRef bytesReturned As Long)
```

Calling Example:

```
Dim buffer (0 to 9) as Byte  
Dim size as Long  
size = UBound(buffer) + 1
```

Call `GetBuffer(buffer(0), size)`

5. Parameters passed as a void pointer (`void*`) should use a value type of Any.

Example:

**C (.h) File:**

```
__declspec(dllexport) void SetObject(void* object);  
__declspec(dllexport) void GetObject(void** object);
```

**VB6 (.bas) File:**

```
Public Declare Sub SetObject Lib "Some.dll" (ByVal object As Any)  
Public Declare Sub GetObject Lib "Some.dll" (ByRef object As Any)
```

Note: Passing pointers can be problematic when dealing with 32-bit/64-bit systems. IntPtr is platform dependent, meaning that it is a four byte pointer on 32-bit systems and an eight byte pointer on a 64-bit system. A .NET application running in 64-bit mode will not be able to load a 32-bit DLL. You must either build a separate 64-bit DLL or modify your .NET project to only run in 32-bit mode.

6. Structures must always be passed by reference in the C DLL and in VB6.

Example:

**C (.h) File:**

```
typedef struct PERSON  
{  
    BYTE id;  
    WORD month;  
    char name[10];  
  
} PERSON, *PPERSON;
```

```
__declspec(dllexport) void GetPerson(PPERSON person);  
__declspec(dllexport) void SetPerson(PPERSON person);
```

**VB6 (.bas) File:**

```
Public Type PERSON  
    id as Byte  
    month as Integer  
    firstName(9) as Byte  
End Type
```

```
Public Declare Sub GetPerson Lib "Some.dll" (ByRef person as PERSON)  
Public Declare Sub SetPerson Lib "Some.dll" (ByRef person as PERSON)
```

Calling Example:

```
Dim person as PERSON
```

```
GetPerson(person)
person.id = 2
Call SetPerson(person)
```

7. Parameters passed as an array of structs should use VB6 references to User Defined Types.

Examples:

**C (.h) File:**

```
typedef struct PERSON
{
    BYTE id;
    WORD month;
    char name[10];
} PERSON, *PPERSON;
```

```
__declspec(dllexport) void GetPeople(PERSON people[], DWORD* numPeople);
__declspec(dllexport) void SetPeople(PERSON people[], DWORD numPeople);
```

**VB.NET (.vb) File:**

```
Public Type PERSON
    id As Byte
    month As Integer
    fname(9) As Byte
End Type
```

```
Public Declare Sub GetPeople Lib "StructTest.dll" (ByRef people As PERSON, ByRef numPeople As Long)
Public Declare Sub SetPeople Lib "StructTest.dll" (ByRef people As PERSON, ByVal numPeople As Long)
```

Calling Example:

```
Dim people(0 To 1) As PERSON
Dim numPeople As Long

numPeople = UBound(people) + 1

people(0).id = 1
people(0).month = 11
people(0).fname(0) = &H31

people(1).id = 2
people(1).month = 12
people(1).fname(1) = &H32
```

Call SetPeople(people(0), numPeople)

Call GetPeople(people(0), numPeople)