

Re: C# – getting binary data from .lib

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.interop/2006-06/msg00045.html>

- *From:* "Michael Phillips, Jr." <mphillips53@xxxxxxxxxxxxxxxx>
 - *Date:* Sun, 4 Jun 2006 10:23:07 -0400
-

Just a few suggestions:

Use a try, catch block as in the following:

```
try
{
Marshal.Copy(ptrImage, managedArray2, 0, 500);
}
catch(ArgumentNullException e)
{
MessageBox.Show( e.Message, "Marshal Copy Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
```

If this throws an exception, then there is a problem with the ptrImage returned by the wrapper.

In your wrapper, check the ptrImage returned from:

```
int retCode = create(id, scale, ptrImage);
```

It should contain 500 bytes of unsigned char data.

Additionally use GlobalSize(*image) after the above call to see if the image array is still 500 bytes.

"Stephen Cawood" <cawood@xxxxxxxx> wrote in message [news:ABvgg.13536\\$\\$61.7798@xxxxxxxx](mailto:news:ABvgg.13536$$61.7798@xxxxxxxx)

thanks a tonne :)

I changed the wrapper to your second suggestion (below) and it seems to be working. the return value is 0 and the pointer has a value.

now I just have to figure out why the copy isn't working.

I tried this...

```
IntPtr ptrImage = Marshal.AllocHGlobal(500);
```

Re: C# – getting binary data from .lib

```
label1.Text = create_wrapped(0, 5, ref ptrImage).ToString();
```

```
byte[] managedArray2 = new byte[500];
```

```
Marshal.Copy(ptrImage, managedArray2, 0, 500);
```

the result is an empty array, but I'm tired so I may be missing something.
cheers

just to be clear, I'm still using:

```
[DllImport("Wrapper.dll", EntryPoint="create_wrapped")]  
public static extern int create_wrapped(int id, int scale, [In][Out] ref  
IntPtr image);
```

"Michael Phillips, Jr." <mphillips53@xxxxxxxxxxxxxxxx> wrote in message
news:uuSAuE1hGHA.4040@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Looking at your original requirements, you have a second choice. You may use the Marshal class to allocate the number of bytes that you need for the image array.

Use Marshal.AllocHGlobal to allocate the memory and pass this IntPtr to the wrapper as I described in my previous post.

You must change your wrapper to the following:

```
WIN32DLL_API int create_wrapped(int id, int scale, void** image)  
{  
    unsigned char* ptrImage = (unsigned char*)GlobalLock(*image);  
    int retCode = create(id, scale, ptrImage);  
    GlobalUnlock(*image);  
  
    return retCode;  
}
```

You now control the pointer returned. Use the Marshal class to read or copy the bytes and use Marshal.FreeHGlobal to free the memory for the unmanaged pointer.

To make things more general you can test to see if the pointer is null to begin with and use one wrapper that will work with unallocated memory or allocate memory as follows:

```
WIN32DLL_API int create_wrapped(int id, int scale, void** image)  
{  
    int retCode = -1;  
    unsigned char* ptrImage = NULL;  
  
    if ( NULL != *image )
```

Re: C# – getting binary data from .lib

```
{  
ptrImage = (unsigned char*)GlobalLock(*image);  
retCode = create(id, scale, ptrImage);  
GlobalUnlock(*image);  
}  
else  
{  
retCode = create(id, scale, ptrImage);  
*image = (void*)ptrImage;  
}  
  
return retCode;  
}
```

"Michael Phillips, Jr." <mphillips53@xxxxxxxxxxxxxxxx> wrote in message
news:OqJ0530hGHA.1324@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

```
label1.Text = create_wrapped(0,5,ref  
ptrImage).ToString();
```

The above line is not correct! You cannot assign binary
image data to
Text as the types are not compatible.

If there was no error, then the value of ptrImage will not be
equal to
IntPtr.Zero.

You may use the Marshal class to copy or read the image
bytes to a
managed C# byte array that you allocate.

Remember the wrapper is returning an unmanaged pointer to
an array of
bytes. You need to convert this to a managed pointer of the
same type.

Additionally, you have to worry about disposing of the
memory allocated
for the unmanaged pointer. On the C# side your pointer is
garbage
collected automatically.

"Stephen Cawood" <cawood@xxxxxxxx> wrote in
message
[news:sbmgg.24496\\$JX1.10537@xxxxxxxx](mailto:news:sbmgg.24496$JX1.10537@xxxxxxxx)

Re: C# – getting binary data from .lib

thanks again. I'm reluctant to try and deal with the unmanaged pointer since I'm getting an error back from the function.

```
label1.Text = create_wrapped(0,5,ref ptrImage).ToString();  
// returned -1 meaning that there was a problem
```

any ideas why this is returning a failure message? when I used byte[] image as the argument, I got back success message (0)... and an empty array :)

"Michael Phillips, Jr."
<mphillips53@xxxxxxxxxxxxxxxx> wrote in message
news:ujJsSm0hGHA.412@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

I am sorry that was a typo.
Use (unsigned char*) rather than (char*).

What you are trying to accomplish is to let the wrapper internally call the library to allocate memory for the image when a null pointer is passed as an argument.

The library should return the pointer to this allocated memory. In "C", this is accomplished by passing a pointer to a pointer on the stack.

In C# you are passing the address to a null pointer. You should get back an unmanaged pointer with the allocated image.

"Stephen Cawood"
<cawood@xxxxxxxx>
wrote in message

Re: C# – getting binary data from .lib

news:4nlgg.24072\$JX1.915@xxxxxxxxxxxx

BTW – the
binary data
is used to
create
.PGM
image files.

"Stephen
Cawood"
<cawood@xxxxxxxxxxxx>
wrote in
message
news:nhlgg.24033\$JX1.23566@xxxxxxxxxxxx

thanks
again,
I
really
appreciate
the
help.
this
problem
has
been
a
real
pain.
I
tried
your
suggestion,
but
it
didn't
work.
however,
I
feel
that
I
might
be
very
close.

label1.Text
=

Re: C# – getting binary data from .lib

```
create_wrapped(0,5,ref
ptrImage).ToString();
//
returned
-1
meaning
that
there
was
a
problem
```

my
original
wrapper
function
was:

```
WIN32DLL_API
int
create_wrapped(int
id,
int
scale,
unsigned
char
*image)
{
return
create(id,
scale,
image);
}
```

so
I
tried
changing
that
to:

```
WIN32DLL_API
int
create_wrapped(int
id,
int
scale,
void**
image)
{
return
```

Re: C# – getting binary data from .lib

```
create(id,  
scale,  
(unsigned  
char*)*image);  
}
```

your
suggestion
wouldn't
compile:

```
WIN32DLL_API  
int  
create_wrapped(int  
id,  
int  
scale,  
void**  
image)  
{  
return  
create(id,  
scale,  
(char*)*image);  
}
```

the
error
was:
Cannot
convert
argument
3
from
char*
to
unsigned
char*

"Michael
Phillips,
Jr."

<mphillips53@xxxxxxxxxxxxxxxx>

wrote

in

message

news:%23EGjMdzhGHA.1508@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Re: C# – getting binary data from .lib

Your wrapper function should have the argument typed as a pointer to a pointer for this to work correctly.

```
[DllImport("Wrapper.dll",  
EntryPoint="create_wrapped")]  
public  
static  
extern  
int  
create_wrapped(int  
id,  
int  
scale,  
[In][Out]  
ref  
IntPtr  
image);
```

```
WIN32DLL_API  
int  
create_wrapped(int  
id,  
int  
scale,  
void**  
image)  
{  
return  
create(id,  
scale,  
(char**)*image);  
}
```

You

Re: C# – getting binary data from .lib

call
it
as
follows:

```
IntPtr  
ptrImage  
=  
IntPtr.Zero;  
label1.Text  
=  
create_wrapped(0,5,ref  
ptrImage);
```

and
of
course
check
for
a
null
pointer
return.

"Stephen
Cawood"
<cawood@xxxxxxxx>
wrote
in
message
[news:oljgg.23800\\$JX1.19291@xxxxxxxx](mailto:news:oljgg.23800$JX1.19291@xxxxxxxx)

I
thought
that
you
might
be
on
to
something,
but
sadly
it
didn't
work.
thanks
for

Re: C# – getting binary data from .lib

the
help
though.

```
[DllImport("Wrapper.dll",  
EntryPoint="create_wrapped")]  
public  
static  
extern  
int  
create_wrapped(int  
id,  
int  
scale,  
[In][Out]  
ref  
IntPtr  
image);
```

```
IntPtr  
ptrImage  
=  
new  
IntPtr();  
label1.Text  
=  
create_wrapped(0,5,ref  
ptrImage).ToString();  
//this  
crashes  
the  
debugger  
with  
no  
exception  
returned,  
I  
tried  
to  
catch  
it  
but  
that  
didn't  
help
```

"Michael
Phillips,
Jr."
<mphillips53@xxxxxxxxxxxxxxxxxx>

Re: C# – getting binary data from .lib

wrote
in
message

news:uJyWUbyhGHA.4044@xxxxxxxxxxxxx

You
could
try
the
same
argument
semantics
used
by
the
"Gdi32"
function
CreateDIBSection
which
returns
an
array
of
bytes
as
a
pointer
argument.

The
c#
function
uses
[In][Out]
ref
IntPtr
to
represent
the
array
of
bytes
returned
for
the
pointer
argument
passed
to
CreateDIBSection.
see

Re: C# – getting binary data from .lib

example:

```
[DllImport("gdi32.dll")]
static
extern
IntPtr
CreateDIBSection(IntPtr
hdc,
[In]
ref
BITMAPINFO
pbmi,
uint
iUsage,
[In][Out]
ref
IntPtr
ppvBits,
IntPtr
hSection,
uint
dwOffset);
```

"Stephen
Cawood"
<cawood@xxxxxxxxxx>
wrote
in
message
[news:3y7gg.16349\\$JX1.2431@xxxx](mailto:news:3y7gg.16349$JX1.2431@xxxx)

I
posted
this
question
a
few
days
ago,
but
it
hasn't
been
resolved
so
I'm
trying
again
(with
better

Re: C# – getting binary data from .lib

info).
I'm
trying
to
access
a
C++
.lib
from
C#.
I
can
get
an
int
back
fine,
the
problem
I'm
having
is
returning
an
array
of
binary
data.
any
ideas?
thanks
in
advance...

.lib
contains
this
function:

```
/-Call  
create()  
to  
create  
a  
bitmap  
of  
(10*scale)  
x  
(*10*scale)  
bytes  
//create()
```

Re: C# – getting binary data from .lib

```
will
fill
an
unsigned
char
array
with
100*scale*scale
bytes
//
//create
ID=567,
5
pixels/bit,
total
image
will
be
50x50
pixels
//this
function
will
malloc
room
if
image
is
NULL
//returns
-1
if
problem,
0
otherwise
```

```
int
create(int
id,
int
scale,
unsigned
char
*image);
```

```
wrapper
DLL:
```

```
WIN32DLL_API
int
```

Re: C# – getting binary data from .lib

```
create_wrapped(int
id,
int
scale,
unsigned
char
*image)
{
return
create(id,
scale,
image);
}
```

```
[DllImport("Wrapper.dll",
EntryPoint="create_wrapped")
public
static
extern
int
create_wrapped(int
id,
int
scale,
byte[]
image);
```

```
byte[]
bArray
=
new
byte[500];
create_wrapped(0,5,bArray);
//bArray
=
array
of
zeros
```

notes:
I've
also
tried...

```
public
static
extern
int
```

Re: C# – getting binary data from .lib

```
create_wrapped(int
id,
int
scale,
[Out][MarshalAs(UnmanagedType
SizeConst=500)]byte[]
image);
//bArray
=
array
of
zeros

public
static
extern
int
create_wrapped(int
id,
int
scale,
[MarshalAs(UnmanagedType
image);
//bArray
=
array
of
zeros

public
static
extern
int
create_wrapped(int
id,
int
scale,[Out]
IntPtr
image);
//returns
-1
failure
```

Re: C# – getting binary data from .lib