

# Re-exporting functions from dynamically linked DLL

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2005-04/msg01588.html>

---

- *From:* "Kevin Crosbie" <caimhinocrosbai@xxxxxxxx>
  - *Date:* Mon, 25 Apr 2005 13:39:30 +0200
- 

Hi,

I have the following situation:

I have a 3rd party DLL which does not include a LIB file, just a header file with typedef declarations for exported functions (It is intended for dynamic linking)

I need to wrap the DLL to allow for callbacks to my own code (in Lisp), and to do this I need to load the DLL using LoadLibrary and access each function that I need using GetProcAddress.

So for example:

The third party header will say:

```
typedef INT (WINAPI * lp_function)();
```

Then in my c wrapper file:

```
#include "3rdparty.h"
```

```
#ifdef _WIN32
#define DllExport __declspec(dllexport)
#else
#define DllExport
#endif
```

```
HINSTANCE hInstance;
lp_function f1;
```

```
DllExport int initialiseAPI()
{
if(NULL==(hInstance=LoadLibrary("demoapi.dll")))
{
return 1;
}
}
```

```
if(NULL==(f1=(lp_function)GetProcAddress(hInstance,"function")))
{
return 1;
}
```

## Re-exporting functions from dynamically linked DLL

}

}

etc.

The I implement the callback code which in turn calls code on my Lisp side.

So my question is:

I want to re-export these dynamically linked functions. I could easily wrap them:

```
DllExport void wrappedf1 (int arg)
{
  f1(arg);
}
```

but I would prefer to just declare it as exported somehow. If I try something like changing their header file to say:

```
typedef DllExport INT (WINAPI * lp_function)();
```

and my code to declare:

```
DllExport lp_function f1;
```

I get a Segmentation Violation.

Anyone have any ideas?

.

- 
- Prev by Date: [\*timer\*](#)
  - Next by Date: [\*Re: CDialog: 1 picture and 1 text label \(VC6\)\*](#)
  - Previous by thread: [\*timer\*](#)
  - Next by thread: [\*subclassing RichEditCtrl and EN\\_SELCHANGE\*](#)
  - Index(es):
    - ◆ [\*Date\*](#)
    - ◆ [\*Thread\*](#)