

Re: MFC-DLL with CString as parameter?

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2008-05/msg00333.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Thu, 08 May 2008 12:29:02 -0400
-

Note that there is a lot of demand to extend the lexical set of languages to include national characters beyond A-Za-z_.

There is a wonderful essay, I forget who wrote it but it dates back to the early 1930s, about "simplified spelling", where letters like "c" are replaced by either "s" or "k", silent letters are dropped, and so on. As each rule is proposed, it is immediately applied to the text, so that by the end of the essay, the text is almost unreadable, but greatly simplified. A parody of this was written by a Scandanavian programmer, because he points out that letters such as Æ æ Ø and ø are fundamental to the language, and it makes no sense; he poses "what would happen if the letter i or c disappeared from English?" and begins to apply the rules, showing how the spellings degrade. He was in particular complaining about the highly ethnocentric attitude that only the 7-bit ANSI subset is valid (and even languages whose code pages have letters like ä, ö, ü can't use them because they are not in the lexical set of the programming language. When this long-overdue change happens, we will probably have to have a GetProcAddressW call added. But for compatibility, there probably will not be a GetProcAddressA because of overload rules.

joe

On Thu, 8 May 2008 18:06:25 +0200, "Giovanni Dicanio" <giovanni.dicanio@xxxxxxxxxxxxx> wrote:

"Joseph M. Newcomer" <newcomer@xxxxxxxxxxxxx> ha scritto nel messaggio news:hq6624509f4g18951orr38hrkkfi3scnkr@xxxxxxxxxxxxx

```
pFunc = (MyFuncType)GetProcAddress(hMod, "myfunc"); // not _T()
```

[...]

but for this one, using the _T() notation would generate an error when you did a Unicode compilation. I like to make this explicit.

And I think that your are right to point out that.

Re: MFC-DLL with CString as parameter?

I think that ANSI-only version of GetProcAddress is because function names must be US-ASCII only in C/C++, e.g. I think that I cannot name a C/C++ function "perché" (because of non-US character "é")

```
void perché( ... ) // C/C++ error
```

```
void perche(...) // all right
```

Giovanni

Joseph M. Newcomer [MVP]

email: newcomer@xxxxxxxxxxxxx

Web: <http://www.flounder.com>

MVP Tips: http://www.flounder.com/mvp_tips.htm

.