

Re: Are _T() and TEXT() macros equivalent?

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-04/msg01376.html>

- *From:* "Mihai N." <nmihai_year_2000@xxxxxxxxxx>
 - *Date:* Thu, 12 Apr 2007 21:40:02 -0700
-

Oh, please don't misunderstand. Should char be 2 bytes, existing code like this WOULD have to be rewritten, perhaps to

....

My point is for new programs (and even existing programs, with sufficient compiler warnings similar to the spiel that VC2005 puts out concerning strcpy_s, for example), defining char to be 2 bytes makes it easy to:

Now I fail to see the advantage.

Your option:

- The old programs need changes
- For new ones is easy to use char as Unicode character
- The result is not standard
- The result is not cross-platform

Current option:

- The old programs need changes
- For new ones is easy to use wchar_t as Unicode character
- The result is standard
- The result is cross-platform

All this to avoid macros?

Again, existing files would have to be converted, or else code to read/write them have to specifically use one byte chars.

You cannot convert legacy files. Legacy files don't mean "files I have saved a while ago"

It also means any other files out there that are not Unicode UTF-16LE, what is called import/export. Think HTML, XML, text.

It would require rework. But that didn't stop the new things like strcpy_s which breaks existing code as well.

Re: Are _T() and TEXT() macros equivalent?

It does not break existing code. It is just a warning.
And strcpy_s is a non-standard, non-portable extension.
It does not change the functionality of strcpy, which is standard.
Same as _T, _tprintf, etc.

--

Mihai Nita [Microsoft MVP, Windows – SDK]

<http://www.mihai-nita.net>

Replace _year_ with _ to get the real email

.