

Re: CString help

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-06/msg01525.html>

- *From:* "Doug Harrison [MVP]" <dsh@xxxxxxxx>
 - *Date:* Fri, 22 Jun 2007 17:03:03 -0500
-

On Thu, 21 Jun 2007 22:25:43 -0700, "Mihai N." <nmihai_year_2000@xxxxxxxx> wrote:

CString::GetBuffer() can increase efficiency in some situations, but is best avoided, IMHO. You will notice that std::string has no such mechanism.

To be avoided, but still best to have it.
I think it is bad that std::string does not have such a thing.
See my "std::string internal buffer access" note at
<http://www.mihai-nita.net/article.php?artID=20060430a>

Practically speaking, you can do this:

```
const int N = 100;
std::string s;
s.resize(N);
int len = WindowsAPI(&s[0], N);
s.resize(len);
```

Though not guaranteed, I don't know any implementation where this won't work. (I'd really like the non-contiguity, reference-counted, etc options to just go away. People who need those things can't do without them, and the possibility they exist complicates things for everyone else.)

Aside: MSDN has gotten a lot better about defining the meaning of N in the code above. As a function parameter, it usually indicates the maximum number of characters that may be written to the buffer, including the terminating nul. As a return value, it usually indicates the number of characters actually written, excluding the nul. This is true for functions like GetWindowText, which truncate the returned text as necessary. What does this imply for the return value?

--
Doug Harrison
Visual C++ MVP