

Re: \x and strings

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2005-04/msg00692.html>

- *From:* "Frank Hickman [MVP]" <fhickman_NOSP@xxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 14 Apr 2005 08:35:48 -0400
-

"Martin Richter [MVP]" <martin.richter@xxxxxxxx> wrote in message
[news:d3ljbm\\$02i\\$1@xxxxxxxxxxxxxxxx](mailto:news:d3ljbm$02i$1@xxxxxxxxxxxxxxxx)
> Hallo John Carson!
>
>> "Martin Richter [MVP]" <martin.richter@xxxxxxxx> wrote in message
>> [news:d3laie\\$d1s\\$1@xxxxxxxx](mailto:news:d3laie$d1s$1@xxxxxxxx)
>>
>>>
>>> If it is a non-unicode string
>>> a\x01bc
>>> will do it.
>>>
>>> If it is a unicode string use
>>> a\x001bc
>>
>>
>>
>> Have you tried compiling this? (It doesn't compile for me.)
>
> No. I am sorry about the wrong infos.
>
> I just read the docs. And I thought that the parser stopped after 2 or 3
> chars, because the documentation should explicitly 2 and 3 chars after \x.
> In fact it parses all chars until it finds a character not matching the
> sequence and this causes an overflow.
> So the only way is what John Carson wrote.
>

The escape sequence for embedded bytes is three characters following the \x
so your "unicode" example is the correct way for non-unicode strings,
a\x001bc will work properly for single byte characters.

--
=====
Frank Hickman
Microsoft MVP
NobleSoft, Inc.
=====

Replace the _nosp@m_ with @ to reply.

-
- *Follow-Ups:*
 - ◆ [Re: \x and strings](#)
 - ◇ *From:* Martin Richter [MVP]

 - *References:*
 - ◆ [\x and strings](#)
 - ◇ *From:* lallous
 - ◆ [Re: \x and strings](#)
 - ◇ *From:* Martin Richter [MVP]
 - ◆ [Re: \x and strings](#)
 - ◇ *From:* John Carson
 - ◆ [Re: \x and strings](#)
 - ◇ *From:* Martin Richter [MVP]

 - Prev by Date: [64k limit for C++ structure](#)
 - Next by Date: [Re: \x and strings](#)
 - Previous by thread: [Re: \x and strings](#)
 - Next by thread: [Re: \x and strings](#)
 - Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)