

## Re: outputting Unicode to console

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2005-09/msg00214.htm>

---

- *From:* "Jochen Kalmbach [MVP]" <[nospam-Jochen.Kalmbach@xxxxxxxxx](mailto:nospam-Jochen.Kalmbach@xxxxxxxxx)>
  - *Date:* Tue, 13 Sep 2005 07:13:27 +0200
- 

Hi Henry!

I need to use low-level I/O such as `write()` to print some messages to `stdout`. However, when the program is compiled to use wide chars the printed output is correspondingly wide!

As long as you only want to display ASCII chars, I would not recommend to use UNICODE in console apps...

If you want to display other chars (like chinese) you need to be aware of the following:

- your console must have a appropriate font which can display the chars (glyphs); the normal (default) font is not able to do this.
- if you use functions like "`wprintf/wcout`", it is not possible to display unicode-characters (see: `wprintf/wcout` has no UNICODE support <http://blog.kalmbachnet.de/?postid=23> <http://blog.kalmbachnet.de/?postid=5>)
- if you use native functions like `write`, then this uses internally `WriteFile` with which is also not possible to output unicode characters...
- to output unicode characters you need to use `WriteConsoleW` (<http://msdn.microsoft.com/library/en-us/dllproc/base/writeconsole.asp>)

Do I have to go through the `WideCharToMultiByte()` process for each message or is there a cleaner solution?

If you just want to display ANSI/ASCII-characters this one solution...

An other would be:  
<code>

## Re: outputting Unicode to console

```
wchar_t wbuf[] = L"HI THERE\n";  
printf("%S", wbuf);  
</code>
```

--  
Greetings  
Jochen

My blog about Win32 and .NET  
<http://blog.kalmbachnet.de/>