
Subject: cout, cin etc. question
Posted by [exhu](#) on Fri, 11 Apr 2008 07:19:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Does U++ library contain stdout, stdin, stderr as FileStream classes?

If not, then how to use the U++ classes to do stdin/out via Stream classes in order to use proper charset (e.g. WIDE char under WinXP) because standard library (in case of MSVC) uses ANSI charset for cin, cout?

Thanks in advance!

Subject: Re: cout, cin etc. question
Posted by [mirek](#) on Fri, 11 Apr 2008 12:02:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

exhu wrote on Fri, 11 April 2008 03:19: Does U++ library contain stdout, stdin, stderr as FileStream classes?

Cout()
Cerr()

instead of Cin stream, there is

```
String ReadStdIn();
```

(reads single input line).

Quote:

If not, then how to use the U++ classes to do stdin/out via Stream classes in order to use proper charset (e.g. WIDE char under WinXP) because standard library (in case of MSVC) uses ANSI charset for cin, cout?

Oh, actually, I did not even know that you can do or bother with UNICODE in console...

Mirek

Subject: Re: cout, cin etc. question
Posted by [exhu](#) on Fri, 11 Apr 2008 12:49:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, actually unicode in console is a very rare use case. It is just related to the bug report I

posted about command line arguments being not unicode and I wanted to print it to the console and thus found out that console output is not unicode aware as well So I don't think it is of great importance.

Subject: Re: cout, cin etc. question
Posted by [tvanriper](#) on Wed, 07 May 2008 02:32:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

In Windows, the console can be relatively easily Unicode-aware under an NT-derived OS.

Instead of `std::cout/std::cin/std::cerr`, you'd use `std::wcout/std::wcin/std::wcerr`.

Typically, I like to have some header file with something like the following:

```
#ifndef UNICODE
#define Tcout std::wcout
#define Tcin std::wcin
#define Tcerr std::wcerr
#else // UNICODE
#define Tcout std::cout
#define Tcin std::cin
#define Tcerr std::cerr
#endif // UNICODE
```

In this way, I can use the standard stream classes for working with TCHARs (again, in Windows). I've done this with a number of console applications in the past.

I don't know as much about U++ in these regards, though.
