

Subject: Re: Why Cout() and Cerr() work so different?
Posted by [sergeynikitin](#) on Wed, 05 Aug 2015 06:08:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Under Linux

Subject: Re: Why Cout() and Cerr() work so different?
Posted by [sergeynikitin](#) on Wed, 05 Aug 2015 06:12:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Win exe under wine

<--- Finished, press any key to close the window --->

Subject: Re: Why Cout() and Cerr() work so different?
Posted by [mirek](#) on Wed, 05 Aug 2015 11:05:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

kov_serg wrote on Tue, 04 August 2015 23:43 Why this code outputs different lines of text?
(OS: WinXP SP3 32bit Upp:8760)

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{

}


```

Output:

It is because different code is used for Cerr output. Cout performs character set conversion

(UTF8->console OEM charset), Cerr does not.

The reason is that conversion has some bad properties and it was deemed unlikely that somebody outputs unicode to error output. We might have been wrong...

Mirek

Subject: Re: Why Cout() and Cerr() work so different?
Posted by [kov_serg](#) on Wed, 05 Aug 2015 11:40:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Cerr() outputs error message with directory name. It works fine until directory has russian name. I just not understand why behaviour is different. Should it be same? May be it should be fixed.

ps: Under ubuntu it works fine because terminal has utf-8 encoding.

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
    CoutUTF8(); // this forces behaviour to be same

    Cout()<<"loading " <<fn<<"\n";
    Cerr()<<"problem " <<fn<<"\n";
}
```

Subject: Re: Why Cout() and Cerr() work so different?
Posted by [kov_serg](#) on Wed, 05 Aug 2015 12:05:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

But this does not help. It becomes even worse.

File Attachments

1) [pic002.PNG](#), downloaded 792 times

The screenshot shows the TheIDE interface with the following components:

- Window Title:** CheckStdout - TheIDE - [C:\work\MyApps\CheckStdout\CheckStdout.cpp UTF-8 CRLF] { MyApps }
- Menu Bar:** File Edit Project Build Debug Assist Setup Help
- Toolbar:** Includes navigation and development icons, and a dropdown menu set to 'MSC9 Speed'.
- Project Explorer (Left):** Shows a tree view with 'Core' as the root, containing sub-items: 'plugin/z', '<prj-aux>', '<ide-aux>', '<temp-aux>', and '<meta>'.
- Code Editor (Center):** Displays the source code for 'CheckStdout.cpp'. The code includes a header, a namespace, a function to convert a string to a specific charset, and a main function that uses 'Cout' and 'Cerr' to output messages.
- Output Console (Bottom Right):** Shows the execution output of 'CheckStdout.exe'. The output demonstrates that 'Cout' correctly displays Cyrillic characters, while 'Cerr' incorrectly displays them as garbled text.

```
#include <Core/Core.h>

using namespace Upp;

String C(const String& s) {
    return ToCharset(CHARSET_CP866,s,CHARSET_UTF8);
}

CONSOLE_APP_MAIN
{
    CoutUTF8();

    String fn="C:\\АБВ\\list.txt";

    fprintf(stdout, "std.loading %s\n", ~C(fn));
    fprintf(stderr, "std.problem %s\n", ~C(fn));

    Cout() << "upp.loading " << (fn) << "\n";
    Cerr() << "upp.problem " << C(fn) << "\n";
    Cout() << "upp.loading " << C(fn) << "\n";

    Cout() << "done\n";
}
```

```
C:\apps\dev\upp-8760\out\MyApps\MSC9.Force_Speed\CheckStdout.exe
std.loading C:\АБВ\list.txt
std.problem C:\АБВ\list.txt
upp.loading C:\АБВ\list.txt
upp.problem C:\АБВ\list.txt
upp.loading C:\АБВ\list.txt
--- Finished in <0:00.06>, exitcode: 0 --->
```

Subject: Re: Why Cout() and Cerr() work so different?
Posted by [Mindtraveller](#) on Wed, 12 Aug 2015 10:18:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

I guess charset conversion for Cout/Cerr could be a bad kind of surprise. This is something you don't wait for when use it.

Subject: Re: Why Cout() and Cerr() work so different?
Posted by [mirek](#) on Thu, 20 Aug 2015 06:20:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

To tell the truth, all is also complicated by the fact that in win32, we present itself to system as "non-UNICODE" application (because of now obsolete demand to support Win95/Win98). That means the console runs in OEM charset... So the conversion is UTF8->OEM charset
