

---

Subject: Hack for MSC8 and "newline in constant" errors

Posted by [kfeng](#) on Mon, 09 Jul 2007 11:43:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I'm new to upp, and I almost gave it up after two days, because it wouldn't compile under MSC8 under WinXP...

...until I found a hacky work-around... ...In Core/Language/t.h, search for TFILE and just comment out the line like this:

```
// #include TFILE
```

or wrap it with an #ifndef:

```
#ifndef PLATFORM_WIN32
#include TFILE
#endif
```

Of course, this is ugly - but given the choice of not being able to compile at all and losing Localization support, I volunteer to take the lesser evil.

I hope some expert developers in upp can come up with a more graceful solution than this, but looking through the threads, I'm not the first one to notice this problem...

- Ken

-----  
F:\upp\uppsrc\CtrlLib\CtrlLib.t : warning C4819: The file contains a character that cannot be represented

in the current code page (932). Save the file in Unicode format to prevent data loss

F:\upp\uppsrc\CtrlLib\CtrlLib.t(29) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(41) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(43) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(53) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(65) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(76) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(77) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(89) : error C2001: newline in constant

F:\upp\uppsrc\CtrlLib\CtrlLib.t(91) : error C2001: newline in constant

---

---

Subject: Re: Hack for MSC8 and "newline in constant" errors

Posted by [mirek](#) on Mon, 09 Jul 2007 11:54:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

So far, the only known solution is to use setup the windows to be "english" (it is really weird, but this global setting affects a way how MSC deals with string literals. The problem here seems to be that in Chinese, the compiler expects to see multibyte characters in literals and for whatever reason, they have decided whether these multibyte sequences are correct, even when they do not do anything with them...).

Mirek

---

Subject: Re: Hack for MSC8 and "newline in constant" errors

Posted by [kfeng](#) on Mon, 09 Jul 2007 12:27:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Actually, it's not just Chinese. I suspect any CJK-based Windows system compiling under MSC8 will have problems (MSC71 seems to be ok). At home, I have a Japanese based WinXP and at work, a Japanese based Windows 2000 machine - both fail to compile with the same errors under MSC8.

It messes too many things up to switch to English just to compile upp so I just comment out the Localization.

The percentage of new users who are configured like this is probably not high, but it is definitely a turn-off for new developers. If there is a way to "just get it to build" (using Macros, even?), I would recommend committing it into the release until MS fixes MSC8... ...or at least put it in some "Known Problems" or "Readme" file so developers can use a hack-around.

I don't mean to be nit-picking about this as it looks like a Microsoft bug, but it would be a pity to lose new developers due to a glitch like this... ...the overall upp system is amazingly clean and intuitive (including the IDE - take some getting used to, but it's actually quite good), and I'm discovering new and interesting implementations of the upp GUI every day.

Keep up the good work!

- Ken

---

Subject: Re: Hack for MSC8 and "newline in constant" errors

Posted by [mirek](#) on Mon, 09 Jul 2007 14:11:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Actually, I would be happy to see any progress here.

In fact, I believe that

#pragma setlocale("C")

(or something like this, not 100% sure at the moment)

should solve it, that is why we put it to all .t files. But I am now not sure when this was in fact introduced, perhaps check you have it there...

Mirek

---

---

Subject: Re: Hack for MSC8 and "newline in constant" errors

Posted by [FrankLin](#) on Thu, 17 Jul 2008 14:17:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

hello, all:

I am also new here and have the same problems.

My environment is as following:

OS: Windows Vista (Traditional Chinese version)

Compiler: minGW, MSC8

After some hacking, I found something and it seems the point.

If we used MSC8 to compile upp, we would encounter an error message like this:

C:\Users\FrankLin\ubin\upp\uppsrc\Core/Core.t(37) : error C2001: newline in constant

Core.t is the UTF-8 format and Windows was used to add "EF BB BF" called BOM to every UTF-8 files in order to identify UTF-8 file. The problem is there is no BOM in Core.t.

When I transferred the original Core.t to the new that Windows can identify, the MSC8 can know the format of this file and compile well.

Although the compilation works, there is another warning message like this:

C:\Users\FrankLin\ubin\upp\uppsrc\Core/Core.t(957) : warning C4566: character represented by universal-character-name '\u00E9' cannot be represented in the current code page (950)

This warning is due to the macro itself. For example: in Core.t

we have

and after preprocessor, we got in some where

So the warning seems reasonable.

There is another problem when I added BOM to both Core.t and CtrlLib.t. The MSC8 can work but minGW compiled failed....

It seems minGW didn't accept BOM added UTF-8 files.

any suggestion is appreciated.

---

---

Subject: Re: Hack for MSC8 and "newline in constant" errors

Posted by [mirek](#) on Sat, 19 Jul 2008 07:47:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

FrankLin wrote on Thu, 17 July 2008 10:17hello, all:

I am also new here and have the same problems.

My environment is as following:

OS:Windows Vista (Traditional Chinese version)

Compiler: minGW, MSC8

After some hacking, I found something and it seems the point.

If we used MSC8 to compile upp, we would encounter an error message like this:

C:\Users\FrankLin\ubin\upp\uppsrc\Core/Core.t(37) : error C2001: newline in constant

Core.t is the UTF-8 format and Windows was used to add "EF BB BF" called BOM to every UTF-8 files in order to identify UTF-8 file. The problem is there is no BOM in Core.t.

When I transfered the original Core.t to the new that Windows can identify, the MSC8 can know the format of this file and compile well.

Although the compilation works, there is another warning message like this:

C:\Users\FrankLin\ubin\upp\uppsrc\Core/Core.t(957) : warning C4566: character represented by universal-character-name '\u00E9' cannot be represented in the current code page (950)

This warning is due to the macro itself. For example:in Core.t

we have

and after preprocessor, we got in some where

So the warning seems reasonable.

There is another problem when I added BOM to both Core.t and CtrlLib.t. The MSC8 can work but minGW compiled failed....

It seems minGW didnt accept BOM added UTF-8 files.

any suggestion is appreciated.

Well, but that is only partial solution...

What we really need here is that the compiler stops checking the content of strings... which should be forced by #pragma setlocale("C") but for whatever reason, it does not work...

(Maybe somebody could check with MSC9).

Mirek

---

---

Subject: Re: Hack for MSC8 and "newline in constant" errors

Posted by [yoco](#) on Fri, 16 Oct 2009 16:46:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I tested with MSVC9, and, you know, MS sux The  
#pragma setlocale ("C")  
still don't work.

---

Subject: Re: Hack for MSC8 and "newline in constant" errors

Posted by [mirek](#) on Sun, 18 Oct 2009 19:30:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

yoco wrote on Fri, 16 October 2009 12:46I tested with MSVC9, and, you know, MS sux The  
#pragma setlocale ("C")  
still don't work.

Yep, they just ignore the thing....

Seriously, the only way out I can see is too monstrous - make it part of build process and actually export all .t files into something else.

I am quite sure I do not really want to follow this path. It would break the code too (you would have to include something else than .t file).

Mirek

---