
Subject: Re: Building & using U++ without TheIDE
Posted by [sergei](#) on Thu, 20 Sep 2007 11:29:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

I didn't want to use Unicode, I really don't care how wrong it writes in debug, that's why I initially left SysChrSet and didn't replace with SsystemCharset like everywhere else. But:

```
::GetModuleFileName(NULL, fn, 512);
```

In UNICODE fn has to be const WCHAR*, and it's char*. So I used the WinCE code:

```
wchar wfn[256];  
::GetModuleFileName(NULL, wfn, 512);  
strcpy(fn, FromSysChrSet(wfn));
```

What crashes is this function in Debug.cpp:

```
#ifdef PLATFORM_WIN32  
static void sLogFile(char *fn, const char *app = ".log")  
{  
#ifdef PLATFORM_WINCE  
    wchar wfn[256];  
    ::GetModuleFileName(NULL, wfn, 512);  
    strcpy(fn, FromSysChrSet(wfn));  
#else  
    ::GetModuleFileName(NULL, fn, 512);  
#endif  
    char *e = fn + strlen(fn), *s = e;  
    while(s > fn && *--s != '\\') *s != '.'  
    ;  
    strcpy(*s == '.' ? s : e, app);  
}  
#endif
```

It's called after APP_MAIN finishes execution. It crashes if you change to this (don't define UNICODE or anything else, crashes "out of the box"):

```
#ifdef PLATFORM_WIN32  
static void sLogFile(char *fn, const char *app = ".log")  
{  
#ifdef PLATFORM_WINCE  
    wchar wfn[256];  
    ::GetModuleFileName(NULL, wfn, 512);  
    strcpy(fn, FromSysChrSet(wfn));  
#else  
    wchar wfn[256];  
    ::GetModuleFileNameW(NULL, wfn, 512);  
    FromUnicode(fn, wfn, wstrlen(wfn), CHARSET_DEFAULT);  
#endif  
}
```

```
#endif
char *e = fn + strlen(fn), *s = e;
while(s > fn && *--s != '\\' && *s != '.')
;
strcpy(*s == '.' ? s : e, app);
}
#endif
```

OK. Not "out of the box". Yesterday it crashed, today GDB just gets stuck and doesn't terminate the program. But the problem is there, since in MSVC it also crashes in debug (unhandled exception in vcont.h). It's the same bug/something, in Code::Blocks it also got segmentation fault somewhere in vectors from FromUnicode call. And both times it happened inside here:

```
inline
static CharSetData& s_cset(byte charset)
{
return s_map()[ResolveCharset(charset)];
}
```

Hope that helps.
