
Subject: serious mingw bug?

Posted by [hojtsy](#) on Sun, 01 Jun 2008 15:48:27 GMT

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

I am using U++ 2008.1beta2 with the embedded MinGW 4.3.0 on WinXp sp2.

I think I have found a code generation bug in MinGW for this code:

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

```
struct Test
```

```
{
    Test() { Upp::Cout() << "Test::Test()\n"; }
    ~Test() { Upp::Cout() << "Test::~~Test()\n"; }
};
```

```
struct Worker
```

```
{
    Worker * Fn1()
    {
        bool throwException = false;
        bool enterLoop = true;
        for(Test t; enterLoop;)
        {
            if(throwException) throw int(3);
            Upp::Cout() << "returning this = " << (void *) this << "\n";
            return this;
        }
    }
}
```

```
int i;
```

```
};
```

```
CONSOLE_APP_MAIN
```

```
{
    Worker w;
    Upp::Cout() << "&w = " << (const void *) &w << "\n";
    Worker *w2 = w.Fn1();
    Upp::Cout() << "received = " << (const void *) w2 << "\n";
}
```

output is below. Notice that Fn1 incorrectly returns the null pointer. Both the non-invoked throw and the loop is important: if I remove any of them, the bug doesn't occur. Am I missing something? Wouldn't it be good idea to revert the included MinGW to a stable version?

```
&w = 0x12FF40
```

```
Test::Test()
```

```
returning this = 0x12FF40
```

```
Test::~~Test()
```

```
received = 0x0
```

- hojtsy
