

## U++ - Feature #460

### Avoid the need of String().Cat() with moving << as String method

04/19/2013 08:17 AM - Miroslav Fidler

<b>Status:</b>	Rejected	<b>Start date:</b>	04/19/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Miroslav Fidler	<b>% Done:</b>	100%
<b>Category:</b>	Core	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour
<b>Description</b>			

#### History

##### #1 - 05/24/2013 05:57 AM - Sender Ghost

- File 460\_uppsrc.diff added
- Category set to Core
- Status changed from New to Patch ready
- Assignee set to Miroslav Fidler

If I understood correctly, it must be possible to do the following:

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

```
using namespace Upp;
```

```
CONSOLE_APP_MAIN
```

```
{  
    String data = "Some data";  
    String text =  
#if 1  
        String()  
#else // as well as  
        String().Cat()  
#endif  
    << data << " and some text\n";  
  
    Cout() << text;  
}
```

The corresponding patch included.

##### #2 - 05/25/2013 10:40 AM - Miroslav Fidler

Actually, that is only a partial solution. This should work too:

```
String text = String() << 123;
```

(-> method has to be the template)

Anyway, we have yet to decide whether this is a good idea after all...

**#3 - 05/25/2013 10:40 AM - Miroslav Fidler**

- Assignee changed from Miroslav Fidler to Sender Ghost

**#4 - 05/26/2013 06:18 AM - Sender Ghost**

- File 460\_uppsrc2.diff added
- Assignee changed from Sender Ghost to Miroslav Fidler
- % Done changed from 0 to 100

I created the second version of the patch, which solves the above case.

**#5 - 05/26/2013 07:01 AM - Sender Ghost**

Miroslav Fidler wrote:

Any way, we have yet to decide whether this is a good idea after all...

I think, the methods are equal (because of using almost the same inlined versions).  
The sizes of applications, compiled with (MinGW) GCC and MSC for current and new methods, are the same.

The main purpose is to "help" GCC compiler (where MSC compiler is ok for current and new methods) to understand:

conversion for argument 1 from 'Upp::String' to 'Upp::String&'.

**#6 - 05/26/2013 09:55 AM - Sender Ghost**

- File 460\_conversion.diff added

Also I created the patch for global conversion of **String().Cat()** to **String()** following directories:

bazaar, examples, rainbow, reference, tutorial, upinbox, uppdev, uppsrc, upptst.

**#7 - 05/08/2016 08:17 AM - Miroslav Fidler**

- Status changed from Patch ready to Rejected

Resolved with C++11 &&

Files			
460_uppsrc.diff	1.03 KB	05/24/2013	Sender Ghost
460_uppsrc2.diff	2.71 KB	05/26/2013	Sender Ghost
460_conversion.diff	76.4 KB	05/26/2013	Sender Ghost