
Subject: String::Make - new way how to create a small string fast

Posted by [mirek](#) on Sat, 18 Sep 2021 15:31:11 GMT

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

While optimizing conversion routines, I have found a relative bottleneck when converting small char buffers to String. To solve that, there is now a new static method

```
template <class Maker> static String Make(int alloc, Maker m)
```

Optimized static method for creating Strings. This method creates internal buffer of at least alloc and then invokes lambda m passing the char * pointer to the internal buffer as lambda parameter. Lambda is then supposed to fill the characters to this buffer and return the length of string (which must be <= alloc). For the best performance, alloc should be constant.

Example:

```
String x = String::Make(12, [](char *t) { *t++ = 'X'; return 1; });
```

Subject: Re: String::Make - new way how to create a small string fast

Posted by [jjacksonRIAB](#) on Mon, 14 Feb 2022 14:23:33 GMT

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

Did you mean

```
String x = String::Make(12, [](char* t) { *t++ = 'X'; return 1; });
```

in your example?

Subject: Re: String::Make - new way how to create a small string fast

Posted by [mirek](#) on Mon, 14 Feb 2022 14:43:16 GMT

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

Yes, thank you, corrected.

Mirek

Subject: Re: String::Make - new way how to create a small string fast

Posted by [jjacksonRIAB](#) on Mon, 14 Feb 2022 14:50:24 GMT

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

You still missed the single quotes on the char. :d
