
Subject: Bug in String/Stream or just misleading documentation?

Posted by [Werner](#) on Wed, 06 Feb 2008 17:19:45 GMT

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

String documentation says that String contains an array of characters which is zero-terminated. This suggests that a String cannot contain '\0's.

On the other hand Stream documentation says:

Quote: void Put(const String& s)

Writes String to the stream. String can contain zero characters.

s String to write.

This can either mean that s can indeed contain '\0's or that "s.GetLength() == 0" is valid.

While the latter case seems trivial to me, the first case doesn't work as outputting a String with an appended '\0' swallows the trailing '\0'.

So, what is correct?

No '\0's inside a String and "a_stream.Put("");" is valid?

Or '\0's inside a String are valid and "Put(const String& s)" is buggy?

Werner

Subject: Re: Bug in String/Stream or just misleading documentation?

Posted by [mirek](#) on Wed, 06 Feb 2008 17:26:37 GMT

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

String is zero terminated, but can contain zeroes...

String s; s[s.GetLength()] == 0 is legal operation and is always true.

Mirek

Subject: Re: Bug in String/Stream or just misleading documentation?

Posted by [Werner](#) on Wed, 06 Feb 2008 20:04:33 GMT

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

luzr wrote on Wed, 06 February 2008 18:26 String is zero terminated, but can contain zeroes...

String s; s[s.GetLength()] == 0 is legal operation and is always true.

Thank you for the clarification.

After thinking again I have to admit that there isn't any bug.

When reading the documentation for the first time, I was thinking that

Quote: void Put(const String& s)

Writes String to the the stream. String can contain zero characters.

s String to write.

means "String may be empty". That was what I wanted to express using the pseudo code "s.GetLength() == 0".

My mistake was that I was trying to build a String with its last character being a '\0' by writing

```
String str;  
str = "xxx\0";
```

overlooking that the character sequence is actually an array of char which of course sends the trailing '\0' to hell.

Sorry for the hassle.

Werner