

Status:	Approved	Start date:	03/26/2017
Priority:	Normal	Due date:	
Assignee:	Zbigniew Rebacz	% Done:	0%
Category:	Core	Estimated time:	0.00 hour
Target version:	Release 2017.2	Spent time:	0.00 hour
Description			
<p>Buffer should also offer alternative method (Get) to ~operator that returns raw pointer.</p> <p>The size could also be remembered and we should introduce Count() and IsEmpty() approach. Something like:</p> <pre>dword length = ::GetShortPathName(static_cast&lt;LPCTSTR&gt;(path), nullptr, 0); if(length == 0)     return path; Buffer&lt;char&gt; shortPathBuffer(length);</pre> <p>Could be easily refactor to:</p> <pre>Buffer&lt;char&gt; GetBufferForShortPath(const String&amp; path) {     dword length = ::GetShortPathName(static_cast&lt;LPCTSTR&gt;(path), nullptr, 0);     return Buffer&lt;char&gt; shortPathBuffer(length); }</pre> <p>// The in the code:</p> <pre>Buffer&lt;char&gt; shortPathBuffer = GetBufferForShortPath(path); if (shortPathBuffer.IsEmpty())     return path; else     // Do something with the buffer...</pre>			

History

- #1 - 03/26/2017 02:53 PM - Zbigniew Rebacz
- Description updated
- #2 - 03/26/2017 02:53 PM - Zbigniew Rebacz
- Subject changed from Buffer should offer Get method similar to One's method to Buffer should offer Get method similar to One's method. Counn and IsEmpty should also be added.
- #3 - 03/26/2017 02:58 PM - Zbigniew Rebacz
- Subject changed from Buffer should offer Get method similar to One's method. Counn and IsEmpty should also be added. to Buffer class improvements (Get method that replace ~operator), count and empty methods.
- Description updated

**#4 - 03/26/2017 02:59 PM - Zbigniew Rebacz**

- *Description updated*

**#5 - 04/09/2017 09:59 AM - Miroslav Fidler**

- *Status changed from New to Ready for QA*

- *Assignee changed from Miroslav Fidler to Zbigniew Rebacz*

I disagree about Count. Rest is implemented.

**#6 - 04/09/2017 03:47 PM - Zbigniew Rebacz**

- *Status changed from Ready for QA to Approved*

OK - without count it could be. Approved!