Reconsidering a null-terminated string view

Document #:	D2001R0
Date:	2019-12-28
Project:	Programming Language C++ Library Evolution Incubator

Reply-to:	Elias Kosunen <isocpp@eliaskosunen.com>

TODO: Insert 2001 references to one's heart content

1 Introduction

In early 2019 in Kona, LEWGI saw [P1402R0], titled "std::cstring_view - a C compatible std::string_view adapter". The consensus from that discussion was, that this problem space is not to be pursued further:

> POLL: We should promise more committee time to pursuing a null-terminated string view, knowing that our time is scarce and this will leave less time for other work.

SF	F	N	A	SA
0	4	4	4	3

> Attendance: 16
>
> CONSENSUS: We will not pursue P1402R0 or this problem space.
>
> Source: [P1402-issue]

The author of this paper feels like this decision was ill-adviced, and in light of more evidence, would like to reignite the discussion. For the sake of this paper, this class is called `std::cstring_view`, but it may be subject to change in future revisions.

2 Scope

This paper is proposing adding a new class with similar characteristics to `std::string_view`, with the notable exception of having an invariant of null-termination.

At this time, this paper does not go into more detail in the design of this class, namely the detail whether the class should store the size of the string.

3 References

[P1402-issue] GitHub issue for P1402.
https://wg21.link/p1402/github

[P1402R0] Andrew Tomazos. 2019. std::cstring_view - a C compatible std::string_view adapter.
https://wg21.link/p1402r0