Make integral overloads of `std::to_string` `constexpr`

Document Number:	**P3438 R0**
Date:	2024-10-13
Project:	ISO JTC1/SC22/WG21: Programming Language C++
Reply-to:	Andreas Fertig ⟨ isocpp@andreasfertig.com ⟩
Audience:	LEWG

Contents

1 Introduction

Since C++20, the language offers a **`constexpr`** version of `std::string`. The utility function `std::to_string` that is used to convert integral numbers to a `std::string` is not **`constexpr`**.

With [P2291R3] `to_chars` can be used for a conversion of integral datatypes to a **`char`**$*$ during constant evaluation. However, a correct implementation of `std::to_string` isn't necessarily trivial when it comes to selecting the buffer size—the task below requires more code.

Currently

```cpp
constexpr std::string my_to_string(int v)
{
  // +1 for minus, +1 for digits10
  constexpr size_t bufsize{
    std::numeric_limits<int>::digits10 +
    2};
  char       buf[bufsize];
  const auto res =
    std::to_chars(buf, buf + bufsize, v);

  return std::string(buf, res.ptr);
}

consteval auto
addSuffix(std::string_view suffix)
{
  return std::views::transform(
    [suffix](auto i) {
      return my_to_string(i).append(
        suffix);
    });
}
```

With proposal

```cpp


consteval auto
addSuffix(std::string_view suffix)
{
  return std::views::transform(
    [suffix](auto i) {
      return to_string(i).append(
        suffix);
    });
}
```

The implementation above is derived from libc++.

2 Implementation

This proposal was implemented in a fork of libc++ from the author [GHUPImpl]. No issues were encountered.

In fact, libc++ already used `to_chars` for the conversion. The implementation was moved from `string.cpp` to `string` without any issues.

2.1 Floating Point

Since [P2291R3], which went into C++23 `to_chars`, is usable in a constant expression to convert the value of an integral datatype into a string. The floating point types were deliberately left off at the time, citing difficulties. This paper doesn't change that. However, I would like to note that as [P3391R0] 1.1 does, it might be time to revisit this decision.

2.2 `std::stacktrace`

Since none of the `std::stacktrace` functions are **`constexpr`** and the entire facility is about tracing at run-time, its `to_string` function, and overloads are untouched by this paper.

2.3 `std::bitset`

The `to_string` function of `std::bitset` is **`constexpr`** since C++23.

3 Proposed wording

This wording is base on the working draft [N4988].

Change in **[string.syn]** 23.4.2:

```
// 23.4.5, numeric conversions
float stof(const string& str, size_t* idx = nullptr);
double stod(const string& str, size_t* idx = nullptr);
long double stold(const string& str, size_t* idx = nullptr);
constexpr string to_string(int val);
constexpr string to_string(unsigned val);
constexpr string to_string(long val);
constexpr string to_string(unsigned long val);
constexpr string to_string(long long val);
constexpr string to_string(unsigned long long val);
string to_string(float val);
string to_string(double val);
string to_string(long double val);

float stof(const wstring& str, size_t* idx = nullptr);
double stod(const wstring& str, size_t* idx = nullptr);
long double stold(const wstring& str, size_t* idx = nullptr);
constexpr wstring to_wstring(int val);
constexpr wstring to_wstring(unsigned val);
constexpr wstring to_wstring(long val);
constexpr wstring to_wstring(unsigned long val);
constexpr wstring to_wstring(long long val);
constexpr wstring to_wstring(unsigned long long val);
wstring to_wstring(float val);
wstring to_wstring(double val);
wstring to_wstring(long double val);
```

Change in **[string.conversions]** 23.4.5:

6 *Throws*: `invalid_argument` if `strtof`, `strtod`, or `strtold` reports that no conversion can be performed. Throws `out_of_range` if `strtof`, `strtod`, or `strtold` sets `errno` to `ERANGE` or if the converted value is outside the range of representable values for the return type.

```
constexpr string to_string(int val);
constexpr string to_string(unsigned val);
constexpr string to_string(long val);
constexpr string to_string(unsigned long val);
constexpr string to_string(long long val);
constexpr string to_string(unsigned long long val);
string to_string(float val);
string to_string(double val);
string to_string(long double val);
```

13 *Throws*: `invalid_argument` if `wcstof`, `wcstod`, or `wcstold` reports that no conversion can be

performed. Throws out_of_range if wcstof, wcstod, or wcstold sets errno to ERANGE.

```
constexpr wstring to_wstring(int val);
constexpr wstring to_wstring(unsigned val);
constexpr wstring to_wstring(long val);
constexpr wstring to_wstring(unsigned long val);
constexpr wstring to_wstring(long long val);
constexpr wstring to_wstring(unsigned long long val);
wstring to_wstring(float val);
wstring to_wstring(double val);
wstring to_wstring(long double val);
```

Modify **[version.syn]**

```
#define __cpp_lib_to_string ~~202306L~~YYYYMML // also in <string>
```

Bibliography

[P2291R3] Daniil Goncharov, Alexander Karaev: *"Add Constexpr Modifiers to Functions to_chars and from_chars for Integral Types in <charconv> Header"*, P2291R3, 2021-09-18.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2291r3.pdf

[P3391R0] Barry Revzin: *"constexpr std::format"*, P3391R0, 2024-09-12.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/p3391r0.html

[N4988] Thomas Köppe: *"Working Draft, Standard for Programming Language C++"*, N4988, 2024-08-04.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/n4988.pdf

[GHUPImpl] Andreas Fertig: *"constexpr to_string* implementation on GitHub".
https://github.com/andreasfertig/llvm-project/tree/af-constexprToString