

Writing Regexp 2021-22 / Unicode

Generated by Doxygen 1.9.3

1 Namespace Index	1
1.1 Namespace List	1
2 File Index	3
2.1 File List	3
3 Namespace Documentation	5
3.1 wr22 Namespace Reference	5
3.2 wr22::unicode Namespace Reference	5
3.2.1 Function Documentation	6
3.2.1.1 from_utf8()	6
3.2.1.2 from_utf8_append()	6
3.2.1.3 from_utf8_write()	6
3.2.1.4 to_utf8() [1/2]	7
3.2.1.5 to_utf8() [2/2]	7
3.2.1.6 to_utf8_append() [1/2]	8
3.2.1.7 to_utf8_append() [2/2]	8
3.2.1.8 to_utf8_write() [1/2]	8
3.2.1.9 to_utf8_write() [2/2]	9
4 File Documentation	11
4.1 include/wr22/unicode/conversion.hpp File Reference	11
4.2 conversion.hpp	12
4.3 src/conversion.cpp File Reference	12
Index	15

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

wr22	5
wr22::unicode	5

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

include/wr22/unicode/ conversion.hpp	11
src/ conversion.cpp	12

Chapter 3

Namespace Documentation

3.1 wr22 Namespace Reference

Namespaces

- namespace [unicode](#)

3.2 wr22::unicode Namespace Reference

Functions

- `std::string to_utf8 (const std::u32string_view &string_utf32)`
Convert a u32string or a u32string_view to a string.
- `std::string to_utf8 (char32_t char_utf32)`
Convert a char32_t to a string.
- `void to_utf8_append (std::string &buffer, const std::u32string_view &string_utf32)`
Convert a u32string or a u32string_view to a string.
- `void to_utf8_append (std::string &buffer, char32_t char_utf32)`
Convert a char32_t to a string.
- `std::u32string from_utf8 (const std::string_view &string_utf8)`
Convert a string or a string_view to a u32string.
- `void from_utf8_append (std::u32string &buffer, const std::string_view &string_utf8)`
Convert a string or a string_view to a u32string.
- `template<typename Iter >`
`void to_utf8_write (Iter &output, char32_t char_utf32)`
Convert a char32_t to a string.
- `template<typename Iter >`
`void to_utf8_write (Iter &output, const std::u32string_view &string_utf32)`
Convert a u32string or a u32string_view to a string.
- `template<typename Iter >`
`void from_utf8_write (Iter &output, const std::string_view &string_utf8)`
Convert a string or a string_view to a u32string.

3.2.1 Function Documentation

3.2.1.1 `from_utf8()`

```
std::u32string wr22::unicode::from_utf8 (
    const std::string_view & string_utf8 )
```

Convert a string or a string_view to a u32string.

Exceptions

<code>'boost::locale::utf::conversion_error'</code>	if <code>string_utf8</code> contains invalid UTF-8.
---	---

Usage example:

```
std::string s1 = "Hello world!";
std::u32string s2 = from_utf8(s1);
assert(s2 == U"Hello world!");
```

3.2.1.2 `from_utf8_append()`

```
void wr22::unicode::from_utf8_append (
    std::u32string & buffer,
    const std::string_view & string_utf8 )
```

Convert a string or a string_view to a u32string.

Equivalent to `from_utf8`, but appends the resulting string to an existing one instead of returning a new string, which is slightly more efficient than `buffer += from_utf8(string_utf8)`.

If not sure which function to choose, start with `from_utf8` instead of `from_utf8_append`.

Exceptions

<code>'boost::locale::utf::conversion_error'</code>	if <code>string_utf8</code> contains invalid UTF-8.
---	---

Usage example:

```
std::u32string report = U"Pressure sensor reading is ";
std::string reading = "5 atm";
from_utf8_append(report, reading);
assert(report == U"Pressure sensor reading is 5 atm");
```

3.2.1.3 `from_utf8_write()`

```
template<typename Iter >
void wr22::unicode::from_utf8_write (
```

```
Iter & output,
const std::string_view & string_utf8 )
```

Convert a string or a string_view to a u32string.

Equivalent to `from_utf8`, but writes the resulting string into an output iterator.

If not sure which function to choose, start with `from_utf8` or `from_utf8_append` instead of `from_utf8`↵
`_write`.

Exceptions

<code>'boost::locale::utf::conversion_error'</code>	if <code>string_utf8</code> contains invalid UTF-8.
---	---

Usage example:

```
std::vector<char32_t> codepoints;
std::string hello = "Hello";
from_utf8_write(std::back_inserter(codepoints), hello);
assert(codepoints == {U'H', U'e', U'l', U'l', U'o'});
```

3.2.1.4 to_utf8() [1/2]

```
std::string wr22::unicode::to_utf8 (
    char32_t char_utf32 )
```

Convert a char32_t to a string.

The resulting string encodes this character in UTF-8. A string is returned and not a char, because one char32_t can be encoded by multiple chars.

Usage example:

```
char32_t c1 = U'';
std::string s2 = to_utf8(c1);
std::cout << s2 << std::endl; // Prints ""
```

3.2.1.5 to_utf8() [2/2]

```
std::string wr22::unicode::to_utf8 (
    const std::u32string_view & string_utf32 )
```

Convert a u32string or a u32string_view to a string.

Usage example:

```
std::u32string s1 = U"";
std::string s2 = to_utf8(s1);
std::cout << s2 << std::endl; // Prints ""
```

3.2.1.6 to_utf8_append() [1/2]

```
void wr22::unicode::to_utf8_append (
    std::string & buffer,
    char32_t char_utf32 )
```

Convert a `char32_t` to a string.

The resulting string encodes this character in UTF-8. Equivalent to `to_utf8`, but appends the resulting string to an existing one instead of returning a new string, which is slightly more efficient than `buffer += to_utf8(char_utf32)`.

If not sure which function to choose, start with `to_utf8` instead of `to_utf8_append`.

Usage example:

```
std::string formula = "2 ";
char32_t times = U'Ⓜ';
to_utf8_append(formula, times);
formula += " 2 = 4";
std::cout << formula << std::endl; // Prints "2 Ⓜ 2 = 4"
```

3.2.1.7 to_utf8_append() [2/2]

```
void wr22::unicode::to_utf8_append (
    std::string & buffer,
    const std::u32string_view & string_utf32 )
```

Convert a `u32string` or a `u32string_view` to a string.

Equivalent to `to_utf8`, but appends the resulting string to an existing one instead of returning a new string, which is slightly more efficient than `buffer += to_utf8(string_utf32)`.

If not sure which function to choose, start with `to_utf8` instead of `to_utf8_append`.

Usage example:

```
std::string sentence = "Hello in Russian is ";
std::u32string rus_hello = U"ру";
to_utf8_append(sentence, rus_hello);
std::cout << sentence << std::endl; // Prints "Hello in Russian is ру"
```

3.2.1.8 to_utf8_write() [1/2]

```
template<typename Iter >
void wr22::unicode::to_utf8_write (
    Iter & output,
    char32_t char_utf32 )
```

Convert a `char32_t` to a string.

Equivalent to `to_utf8`, but writes the resulting string to an output iterator instead of returning a new string.

If not sure which function to choose, start with `to_utf8` or `to_utf8_append` instead of `to_utf8_write`.

Usage example:

```
std::cout << "2 ";
char32_t times = U'Ⓜ';
auto iter = std::ostream_iterator<char>(std::cout);
to_utf8_write(iter, times);
std::cout << " 2 = 4";
// Stdout: "2 Ⓜ 2 = 4"
```

3.2.1.9 to_utf8_write() [2/2]

```
template<typename Iter >  
void wr22::unicode::to_utf8_write (   
    Iter & output,  
    const std::u32string_view & string_utf32 )
```

Convert a u32string or a u32string_view to a string.

Equivalent to to_utf8, but writes the resulting string to an output iterator instead of returning a new string.

If not sure which function to choose, start with to_utf8 or to_utf8_append instead of to_utf8_write.

Usage example:

```
std::cout << "Hello in Russian is ";  
std::u32string rus_hello = U"";  
auto iter = std::ostream_iterator<char>(std::cout);  
to_utf8_write(iter, rus_hello);  
// Stdout: "Hello in Russian is "
```


Chapter 4

File Documentation

4.1 include/wr22/unicode/conversion.hpp File Reference

```
#include <string>
#include <string_view>
#include <boost/locale/encoding_utf.hpp>
```

Namespaces

- namespace [wr22](#)
- namespace [wr22::unicode](#)

Functions

- `std::string wr22::unicode::to_utf8 (const std::u32string_view &string_utf32)`
Convert a u32string or a u32string_view to a string.
- `std::string wr22::unicode::to_utf8 (char32_t char_utf32)`
Convert a char32_t to a string.
- `void wr22::unicode::to_utf8_append (std::string &buffer, const std::u32string_view &string_utf32)`
Convert a u32string or a u32string_view to a string.
- `void wr22::unicode::to_utf8_append (std::string &buffer, char32_t char_utf32)`
Convert a char32_t to a string.
- `template<typename Iter >`
`void wr22::unicode::to_utf8_write (Iter &output, char32_t char_utf32)`
Convert a char32_t to a string.
- `template<typename Iter >`
`void wr22::unicode::to_utf8_write (Iter &output, const std::u32string_view &string_utf32)`
Convert a u32string or a u32string_view to a string.
- `std::u32string wr22::unicode::from_utf8 (const std::string_view &string_utf8)`
Convert a string or a string_view to a u32string.
- `void wr22::unicode::from_utf8_append (std::u32string &buffer, const std::string_view &string_utf8)`
Convert a string or a string_view to a u32string.
- `template<typename Iter >`
`void wr22::unicode::from_utf8_write (Iter &output, const std::string_view &string_utf8)`
Convert a string or a string_view to a u32string.

4.2 conversion.hpp

[Go to the documentation of this file.](#)

```

1 #pragma once
2
3 // stl
4 #include <string>
5 #include <string_view>
6
7 // boost
8 #include <boost/locale/encoding_utf.hpp>
9
10 namespace wr22::unicode {
11
12     std::string to_utf8(const std::u32string_view& string_utf32);
13
14     std::string to_utf8(char32_t char_utf32);
15
16     void to_utf8_append(std::string& buffer, const std::u32string_view& string_utf32);
17
18     void to_utf8_append(std::string& buffer, char32_t char_utf32);
19
20     template <typename Iter>
21     void to_utf8_write(Iter& output, char32_t char_utf32) {
22         auto codepoint = static_cast<boost::locale::utf::code_point>(char_utf32);
23         boost::locale::utf::utf_traits<char>::encode(codepoint, output);
24     }
25
26     template <typename Iter>
27     void to_utf8_write(Iter& output, const std::u32string_view& string_utf32) {
28         for (auto char_utf32 : string_utf32) {
29             to_utf8_write(output, char_utf32);
30         }
31     }
32
33     std::u32string from_utf8(const std::string_view& string_utf8);
34
35     void from_utf8_append(std::u32string& buffer, const std::string_view& string_utf8);
36
37     template <typename Iter>
38     void from_utf8_write(Iter& output, const std::string_view& string_utf8) {
39         auto it = string_utf8.begin();
40         auto end = string_utf8.end();
41         while (it != end) {
42             auto codepoint = boost::locale::utf::utf_traits<char>::decode(it, string_utf8.end());
43             if (!boost::locale::utf::is_valid_codepoint(codepoint)) {
44                 throw boost::locale::conv::conversion_error{};
45             }
46             auto char_utf32 = static_cast<char32_t>(codepoint);
47             *output = char_utf32;
48             ++output;
49         }
50     }
51 }
52
53 // namespace wr22::unicode

```

4.3 src/conversion.cpp File Reference

```

#include <boost/locale/encoding_errors.hpp>
#include <boost/locale/utf.hpp>
#include <wr22/unicode/conversion.hpp>
#include <iterator>
#include <boost/locale/encoding_utf.hpp>

```

Namespaces

- namespace [wr22](#)
- namespace [wr22::unicode](#)

Functions

- `std::string wr22::unicode::to_utf8` (const std::u32string_view &string_utf32)
Convert a u32string or a u32string_view to a string.
- `std::string wr22::unicode::to_utf8` (char32_t char_utf32)
Convert a char32_t to a string.
- `void wr22::unicode::to_utf8_append` (std::string &buffer, const std::u32string_view &string_utf32)
Convert a u32string or a u32string_view to a string.
- `void wr22::unicode::to_utf8_append` (std::string &buffer, char32_t char_utf32)
Convert a char32_t to a string.
- `std::u32string wr22::unicode::from_utf8` (const std::string_view &string_utf8)
Convert a string or a string_view to a u32string.
- `void wr22::unicode::from_utf8_append` (std::u32string &buffer, const std::string_view &string_utf8)
Convert a string or a string_view to a u32string.

Index

- from_utf8
 - wr22::unicode, [6](#)
- from_utf8_append
 - wr22::unicode, [6](#)
- from_utf8_write
 - wr22::unicode, [6](#)
- include/wr22/unicode/conversion.hpp, [11](#), [12](#)
- src/conversion.cpp, [12](#)
- to_utf8
 - wr22::unicode, [7](#)
- to_utf8_append
 - wr22::unicode, [7](#), [8](#)
- to_utf8_write
 - wr22::unicode, [8](#)
- wr22, [5](#)
- wr22::unicode, [5](#)
 - from_utf8, [6](#)
 - from_utf8_append, [6](#)
 - from_utf8_write, [6](#)
 - to_utf8, [7](#)
 - to_utf8_append, [7](#), [8](#)
 - to_utf8_write, [8](#)