1 Synopsis

C++ 17 library providing Byte order detection. The library is made available publicly on Github under the MIT License.

2 Limitations and Restrictions

The library officially only supports Visual Studio 2017 and Windows 10. The Byte Order for Windows 10 is always Little Endian.

3 Introductory example

To determine target environment's Byte order, evaluate the constant expression function get_byte_order‘ which returns an enum class element from the byte_order enum class.

```
#include "primordialmachine/byte_order/include.hpp"
#include <stdlib.h>

int main(int argc, char **argv) {
  using namespace primordialmachine;
  using namespace std;
  switch (get_byte_order()) {
    case byte_order::little_endian:
      {
        cout << "Environment Byte order is Little Endian"
             << std::endl;
        return EXIT_SUCCESS;
      } break;
    case byte_order::big_endian:
      {
        cout << "Environment Byte order is Big Endian"
             << std::endl;
        return EXIT_SUCCESS;
      } break;
    default:
     {
        // This code should never be reached.
        err << "unable to determine Environment Byte order"
            << std::endl;
        return EXIT_FAILURE;
     }
  };
}
```

If the Byte order can not be determined, a compile-time error is raised i.e. the program will not compile.

4 Building under Visual Studio 2017

1. Open the solution `byte-order.sln` in Microsoft Visual Studio 2017.
2. Batch build everything.
3. The folder `packages` contains the distribution of the library i.e. include files and the static libraries for
   (a) the platforms `Win32` and `x64` and
   (b) configurations `Release` and `Debug`.
4. Copy the contents of the ‘packages‘ folder into a directory. Let `[library home]` be a placeholder denoting the path by which that folder can be referenced from your project.
5. Add

(a) the include path

```
[library home]/primordialmachine/byte-order/
$(Platform.toLower())/$(Configuration.toLower())/includes
```

and

(b) the library path

```
[library home]/primordialmachine/byte-order/
$(Platform.toLower())/$(Configuration.toLower())/libraries
```

to your project.

6. Link your project with the library `byte-order.lib`.
7. Add the include directive **#include** "primordialmachine/byte_order/include.hpp" where appropriate.
8. You can now use the functionality provided by the library.

5 Library Interface Documentation

5.1 namespace primordialmachine

The namespace this library is adding its declarations/definitions to. The added namespace elements are documented below.

5.2 enum class byte_order

An enum class of the Byte orders known to this library. The contained enum class elements are documented below.

1. little_endian enum class element denoting Little Endian Byte order.
2. big_endian enum class element denoting Big Endian Byte order.

5.3 byte_order get_byte_order()

A parameterless constant expression function getting the target environment's Byte order. It returns a value of type byte_order denoting the target environment's Byte order. If the Byte order can not be determined, a compile-time error is raised i.e. the program will not compile.