Contents

1 The `Endian` Class

The `Endian` Class is a simple little class for testing endianess on a system. Its used primarily (for this project) in determining the placement of HI and LO digits of number data. For example an `uint32_t` has a size of four bytes, and the first two and latter two bytes are each a digit of type `uint16_t`. Which is the Hi or Lo digit? Use the `Endian` class to find out.

1.1 Using `Endian`

To use `Endian` just make use of its members directly as seen in Example 1.

Example 1: Testing Endianess

```
1  uint32_t num = 1048833;
2  // 1048833 == [0000 0000 0001 0000][0000 0001 0000 0001]
3  uint16_t* numLoPart = nullptr;
4  if(cg::Endian::litte)
5    /*system is little endian*/
6    numLoPart = (uint16_t) &num;
7  else
8    /*system is big endian*/
9    numLoPart = ((uint16_t) &num) + 1;  //Move to the first uint16_t
       data plus 1 unit of uint16_t in the significance direction.
10 *numLoPart = 0; // num == 1048576
11 // 1048576 == [0000 0000 0001 0000][0000 0000 0000 0000]
```

1.2 Final Thoughts

A great example is in the source code for the `Num<T>` class header. The member functions use the endian class to detect system endianess to properly decompose larger data types into multiple smaller data types as a reference or copy.