dhorn::utf8\_string

# Introduction

A significant percentage of the web deals not with “c-style” ANSI strings, or with “wide character” Unicode strings, but with UTF-8 encoded strings. UTF-8 encoded strings provide numerous benefits, especially on the web for numerous reasons including, but not limited to:

1. It significantly reduces the size of transmitted data since the percentage of characters that can be transmitted as a single byte is significantly high (and three or four bytes are rarely needed).
2. UTF-8 encoded strings are independent of processor architecture (big vs. little endian) since strings are represented as a sequence of bytes (the smallest unit of transmission).
3. Many files (.xml files in particular) are moving to a UTF-8 encoded format.

The negative side of this is that many C++ applications (and the STL itself) are built around either ANSI or Unicode strings. This makes it increasingly hard for new developers and new code to adopt to the web in C++ applications. The “solution” that many developers and companies have taken is to move to alternate languages, mainly scripting languages, which either deal with UTF-8 encoded strings by default or do any conversion automatically. This is not to say that nobody has tried to bring UTF-8 encoded strings to the C++ language as there are likely hundreds of “reasonable” implementations you can find by browsing around.

# Classes/Structures/Types/Enumerations/etc.