Subject: Reliably converting between std::vector<uint8_t> and WString
Posted by steveo on Sat, 21 Nov 2020 23:17:33 GMT
View Forum Message <> Reply to Message

I have a desire to use some encryption libraries which are not WString aware.

How do I reliably take the raw buffer of a WString and convert it to a std::vector<uint8_t>?

and conversely.

How do I reliably take a buffer thus converted and reliably put it back into a WString?

It's been years since I did pointer arithmetic and back in the day we simply ignored WString because we could.

```
WString input = "Some unicode text";
std::vector<uint8_t> bytes(const uint8_t*)input.Begin(), input.GetCount()*sizeof(wchar));
WString output((const wchar*)bytes.data(), (bytes.size()/sizeof(wchar));
```

Subject: Re: Reliably converting between std::vector<uint8_t> and WString
Posted by Oblivion on Sat, 21 Nov 2020 23:26:46 GMT
View Forum Message <> Reply to Message

> Quote:How do I reliably take the raw buffer of a WString and convert it to a std::vector<uint8_t>?

Both Upp::WString and Upp::String classes have a ToStd() method and relevant constructors which converts them to std::string or from std::string to them, if that's what you need.

If you simply need Utf16 to Utf8 conversion, there is also a ToUtf8 method that takes a WString and converts it to Utf8.

Best regards,
Oblivion