The amazing disappearing, reappearing trigraphs

A conspiracy theory too big to fail, by

Paul “TBBle” Hampson

2009: The United Kingdom suggests deprecating trigraphs for C++0x

N2837: C++0X, CD 1, National Body Comments (UK 11)

2009: Canada (and IBM) find this vexing

N2910: Comment on Proposed Trigraph Deprecation

2010: CWG splits the difference, and disables trigraphs in raw string literals

N3077: Alternative approach to Raw String issues

2014: Richard Smith proposes removing trigraphs completely from C++17

N3981: Removing trigraphs??!

2014: Trigraphs removed from C++17

N4053: WG21 2014-06 Rapperswil Minutes

2014: Richard Smith becomes C++ Project Editor

N4053: WG21 2014-06 Rapperswil Minutes

2014: IBM's vexation is more emphatic:
"C++ is becoming an ASCII-dominated language where the annoyances of the many should outweigh the needs of the few"

N4210: IBM comment on preparing for a Trigraph-adverse future in C++17

2014: EWG agrees to support u8 character literals and requests wording

EWG Issue 119

2014: u8 character literals are added to C++17, explicitly limited to US-ASCII

N4197: Adding u8 character literals

2017: `operator<=>` proposed

P0515R0: Consistent comparison

`operator<=>`

P0515R0: Consistent comparison

operator🚀

Unicode Character 'ROCKET' (U+1F680)

```
operator\U0001F680
```

Unicode Character 'ROCKET' (U+1F680)

operator⇔

operator⇔

Unicode Character 'LEFT RIGHT DOUBLE ARROW' (U+21D4)

```
operator\u21D4
```

Unicode Character 'LEFT RIGHT DOUBLE ARROW' (U+21D4)

The truth is out there

https://wg21.link/

??=ASCIIlluminati