C++ IDENTIFIERS USING UAX 31

STEVE DOWNEY

Created: 2020-09-23 Wed 13:18

TABLE OF CONTENTS

- C++ Identifier Syntax using Unicode Standard Annex 31
- The Emoji Problem
- Script Issues
- Other adopters
- We have wording

C++ IDENTIFIER SYNTAX USING UNICODE STANDARD ANNEX 31

- That C++ identifiers match the pattern

(XID_Start + _) + XID_Continue.*

- That portable source is required to be normalized as NFC.
- That using unassigned code points be ill-formed.

PROBLEM THIS FIXES : NL 029

Allowed characters include those from U+200b until U+206x; these are zero-width and control characters that lead to impossible to type names, indistinguishable names and unusable code & compile errors (such as those accidentally including RTL modifiers).

OTHER "WEIRD IDENTIFIER CODE POINTS"

- The middle dot · which looks like an operator.
- Many non-combining "modifiers" and accent marks, such as ´ and ¨ and ., which don't really make sense on their own.
- "Tone marks" from various languages, including ˫ (similar to a box-drawing character ├ which is an operator).
- The "Greek question mark" ; (see below)
- Symbols which are simply not linguistic, such as ۞ and ༒.

https://gist.github.com/jtbandes/c0b0c072181dcd22c3147802025d0b59#weird-identifier-code-points

UAX 31 - UNICODE IDENTIFIER AND PATTERN SYNTAX

- Follows the same principles as originally used for C++
- Actively maintained
- Stable

XID_START AND XID_CONTINUE

- Unicode database defined properties
- Closed under normalization for all four forms
- Once a code point has the property it is never removed
- Roughly:
  - Start == letters
  - Continue == Start + numbers + some punctuation

THE EMOJI PROBLEM

- The emoji-like code points that we knew about were excluded
- We included all unassigned code points
- Emoji 'support' is an accident, incomplete, and broken

SOME EXAMPLES

```
int ⏰ = 0; //not valid
int 🕐 = 0; // valid

int ☠ = 0; //not valid
int 💀 = 0; // valid

int ✋ = 0; //not valid
int 👊 = 0; // valid

int ✈ = 0; //not valid
int 🚀 = 0; // valid

int ☹ = 0; //not valid
int 😀 = 0; // valid
```

♀ AND ♂ ARE DISALLOWED

```
// Valid
    bool 👷 = true; //  Construction Worker
// Not valid
    bool 👷‍♀️ = false; // Woman Construction Worker ({Construction Worker}{ZWJ}{Female Sign})
```

EMOJI ARE NOT "STABLE" IN UNICODE

From the emoji spec

> *isEmoji(♟)=false for Emoji Version 5.0, but true for Version 11.0.*

It is possible that the emoji property could be removed.

SOME SURPRISING THINGS ARE EMOJI

```
002A          ; Emoji              # E0.0   [1] (*)       asterisk
0030..0039    ; Emoji              # E0.0  [10] (0..9)    digit zero..digit nine
```

```
   {DIGIT ONE}{VARIATION SELECTOR-16}{COMBINING ENCLOSING KEYCAP} 1⃣

   {ASTERISK}{VARIATION SELECTOR-16}{COMBINING ENCLOSING KEYCAP} *⃣
```

FIXING THE EMOJI PROBLEM WOULD MEAN BEING INVENTIVE

Being inventive in an area outside our expertise is HARD

Adopting UAX31 as a base to move forward is conservative

SCRIPT ISSUES

Some scripts require characters to control display or require punctuation that are not in the identifier set.

THIS INCLUDES ENGLISH

- Apostrophe and dash
  - Won't, Can't, Mustn't
  - Mother-in-law
- Programmers are used to this and do not notice

ZWJ AND ZWNJ

Zero width joiner and non joiners are used in some scripts

- Farsi word "names"

```
نامهای
NOON + ALEF + MEEM + HEH + ALEF + FARSI YEH
```

نامهای

- Farsi word "a letter"

```
نامه‌ای
NOON + ALEF + MEEM + HEH + ZWNJ + ALEF + FARSI YEH
```

نامه‌ای

Anecdotally, these issues are understood and worked around

OTHER ADOPTERS

- Java (https://docs.oracle.com/javase/specs/jls/se15/html/jls-3.html#jls-3.8)
- Python 3 https://www.python.org/dev/peps/pep-3131/
- Erlang https://www.erlang.org/erlang-enhancement-proposals/eep-0040.html
- Rust https://rust-lang.github.io/rfcs/2457-non-ascii-idents.html
- JS https://tc39.es/ecma262/

WE HAVE WORDING

Core change

identifier:

~~identifier-nondigit~~ identifier-start

identifier ~~identifier-nondigit~~ identifier-continue

~~identifier digit~~

identifier-start:

nondigit

universal-character-name of class XID_Start

identifier-continue:

digit

nondigit

universal-character-name of class XID_Continue