J16-97-0074 = WG21/N1112 (also WG14/N770)
Trigraphs and Universal Character Names
Randy Meyers
23 Sept 1997

Both C and C++ have adopted the same proposal for handling the full range of natural language characters in source programs. Basically, during phase 1 of translation any character not in the basic source character set is mapped into its Universal Character Name (UCN). After phase 1, C/C++ programs are represented only using basic source characters and UCNs.

Phase 1 of translation also handles another mapping: it recognizes trigraphs and translates them into their single character representation. This raises an ordering problem: if the initial source character set is multibyte, do you recognize trigraphs before or after recognizing multibyte characters?

Consider phase 1 input that looks like this:

```
        $??)
```

where in the multibyte encoding, a byte containing the code for "$" is the first byte of a single multibyte character made from the byte containing the "$" and the byte that follows it. Bytes containing the codes for "?" and ")" are treated as single byte characters unless immediately preceded by a special flag byte like "$".

If you process trigraphs before decoding multibyte characters, you would recognize the trigraph for "]", and map the input into "$]", which would then be translated into a surprising multibyte character. The translator would interpret the source completely differently than any display hardware or text processing program.

The alternative, of course, is to perform multibyte processing before trigraph recognition. In that case, the source would be interpreted the same way that the programmer's editor probably displayed it: a multibyte character followed by the characters "?" and ")". This is clearly the most reasonable interpretation, and it also is the most defensible interpretation since phase 1 in the Working Paper talks about recognizing and mapping characters, and trigraph sequences are defined to be sequences of characters. A byte stream before multibyte processing is not a sequence of characters, and you can not find trigraph sequences in it until you turn it into characters by multibyte processing.

Unfortunately, the wording for Phase 1 (Subclause 2.1) in the Post-London Preview Edition of the C++ Working Paper is very easy to misread as requiring trigraph processing before multibyte processing:

1. Physical source file characters are mapped, in an implementation-defined manner, to the basic source character set (including new-line characters for end-of-line

indicators) if necessary. Trigraph sequences (2.3) are replaced by corresponding single-character internal representations. Any source file character not in the basic source character set (2.2) is replaced by the universal-character-name that designates that character. (An implementation may use any internal encoding, so long as an actual extended character in the source file, and the same extended character expressed in the source file as a universal-character-name (i.e., using the notation), are handled equivalently.)

(The wording in the C Working Paper is not yet available, but is expected to be the similar.)

Since the above wording discussing trigraph processing before UCN processing, it appears that trigraph processing happens first.

A simple reordering of the paragraph seems sufficient to clear this problem up:

1. Physical source file characters are mapped, in an implementation-defined manner, to the basic source character set (including new-line characters for end-of-line indicators) if necessary. Any source file character not in the basic source character set (2.2) is replaced by the universal-character-name that designates that character. (An implementation may use any internal encoding, so long as an actual extended character in the source file, and the same extended character expressed in the source file as a universal-character-name (i.e., using the notation), are handled equivalently.) Trigraph sequences (2.3) are replaced by corresponding single-character internal representations.

If the committee wishes, the word "Then" could be inserted at the start of the last sentence to add more emphasis:

> Then, trigraph sequences (2.3) are replaced by corresponding single-character internal representations.

Both the C and C++ Working Papers should reorder the paragraph for clarity and optionally add the word "Then".