Doc. no.:	X3J16 / 95-0220
	WG21 / N0820
Date:	22 November 1995
Project:	Programming Language C++
Reply to:	Dag Brück
	E-mail: `dag@dynasim.se`

Extended characters in C++ programs

Tom Plum and Dag Brück

1. Purpose

The purpose of this proposal is to address CD Ballot comments from France and Sweden, and also long-standing Japanese requirements. The basic elements are:

- An extended character repetoire for identifiers and literals.
- A trigraph encoding of all characters in ISO 10646.

For more background information, see Tom Plum's document SC22 / WG21 N0808R1.

2. Proposal

The implementation shall accept source files encoded in ISO Latin-1. The implementation shall also accept source files using all characters representable in ISO 10646. The source need not be encoded according to ISO 10646, but the implementation must provide "escape sequences" to represent characters in ISO 10646 that are not available in the source character set.

The proposal also defines trigraphs for characters in ISO 10646 that cannot be represented in the source character set. ISO / IEC JTC1 / SC22 has endorsed the use of the ISO 10646 4-digit hex (or 8-digit hex) as a "short-name" for a character. For example, the short-name for "LATIN SMALL LETTER U WITH DIARESIS" would be something like u00FC. (The letter "U" is an explicit part of the short-name.) The short-name is not (necessarily) the target character-set code for the character that it names; the character u00FC presumably has an EBCDIC code which does not equal 00FC. The short-name simply names a specific character.

Programming language standards should allow characters outside the repertoire of the latin nonaccented upper and lower case letters in identifiers. The repertoire recommended consists of characters that are used to write different natural languages of the world, and consists of letters, ideographics and syllabic characters.

ISO / IEC PDTR 10176 presents a set of "International identifier characters," specifically a list of characters of UCS-2 (Unicode) which are proposed for acceptance as "extended identifier" characters. A conforming ISO C++ translator shall accept any of the extended identifier characters in identifiers.

Each character present in a character literal or a string literal shall designate an implementation-defined character in the target character set; otherwise, the program is ill-formed. The trigraph syntax provides a portable syntax for defining characters which are not necessarily present in each target environment.

3. Changes to working paper

Paragraph 1 of [lex.phases] is amended with:

> The implementation shall accept source files encoded in ISO Latin-1. The implementation shall accept source files using all characters representable in ISO 10646. [Note: The source need not be encoded according to ISO 10646, but the implementation must provide "escape sequences" to represent characters in ISO 10646 that are not available in the source character set.]

Paragraph 1 of [lex.trigraph] is amended with:

> A trigraph of the form
>
> `??uxxxx` or `??Uyyyyyyyy`
>
> is replaced by a representation of the character that corresponds to the ISO 10646 canonical short name `xxxx` or `yyyyyyyy`. [Note: we use ISO 10646 to name the extended characters encoded with the `??u` and `??U` trigraphs; this does not imply any particular encoding.]
>
> [Example: `??u00FC` is replaced with the character known as "Latin small letter U with diaresis" in ISO 10646.]

Paragraph 1 of [lex.name] is amended with:

```
nondigit:   extended-character
            <same as current WP>
```

> Extended characters are defined in ISO / IEC PDTR 10176 (Appendix A: Extended repertoire for identifiers).
>
> [Box: The reference to PDTR 10176 must be double-checked.]

Strictly speaking, the list of letters are already part of the extended characters listed in PDTR 10176, but the presentation is clearer if we keep them anyway.