WG21/N0886
X3J16/96-0068
1996-03-13

Extended Identifiers and Extended Literals

Thomas Plum, John Benito, Clark Nelson

Move that we revise the Working Paper as follows:

Item 1) In 2.1, Phases of Translation, add a new paragraph 1 to precede the existing first paragraph. Then add onto "phase 1" a new sentence, "Any source file character not in the basic source character set is replaced by the _universal-character-name_ that designates that character." so that the first two paragraphs would read as follows:

2.1 Phases of translation [lex.phases]

1 The _basic source character set_ consists of 96 characters: the space character, the control characters representing horizontal tab, vertical tab, form feed, and new-line, plus the following 91 graphical characters:

```
  a b c d e f g h i j k l m n o p q r s t u v w x y z        (26)
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z        (26)
  0 1 2 3 4 5 6 7 8 9                                       (10)
  _ { } [ ] # ( ) < > % : ; . ? * + - / ^ & | ~ ! = , \ " ’   (29)
```

The _universal-character-name_ construct provides a way to name other characters. The character designated by the _universal-character-name_ ??UNNNNNNNN is that character whose encoding in ISO/IEC 10646 is the hexadecimal value NNNNNNNN; the character designated by the _universal-character-name_ ??uNNNN is that character whose encoding in ISO/IEC 10646 is the hexadecimal value 0000NNNN.

```
hex-quad:
   hexadecimal-digit hexadecimal-digit hexadecimal-digit hexadecimal-digit

universal-character-name:
   ??u hex-quad
   ??U hex-quad hex-quad
```

2 The precedence among the syntax rules of translation is specified by the following phases.

1 Physical source file characters are mapped to the source character set (introducing new-line characters for end-of-line indicators) if necessary. Trigraph sequences (2.2) are replaced by corresponding single-character internal representations. Any source file character not in the basic source character set is replaced by the _universal-character-name_ that designates that character. [Footnote -- The process of handling extended characters is specified in terms of mapping to an encoding that uses only the basic source character set, and, in the case of character literals and strings, further mapping to the execution character set. In practical terms, however, any internal encoding may be used, so long as an actual extended character encountered in the input, and the same extended character expressed in the input as an _universal-character-name_ (i.e. using the ??uXXXX notation), are handled equivalently.]

[end of quote from revised WP]

Item 2) In 2.1, Phases of Translation, revise "phase 5" as follows:

5 Each source character set member, escape sequence, or _universal-character-name_ in character literals and string literals is converted to a member of the execution character set.

Item 3) In 2.8, Identifiers, add a new line into the definition of _nondigit_, and modify paragraph 1, so that the revised text of 2.8 reads as follows:

2.8 Identifiers [lex.name]

```
identifier:
   nondigit
   identifier nondigit
   identifier digit

nondigit: one of
   _universal-character-name_
   _ a b c d e f g h i j k l m n o p q r s t u v w x y z
     A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

digit: one of
     0 1 2 3 4 5 6 7 8 9
```

1 An identifier is an arbitrarily long sequence of nondigits and digits. Each _universal-character-name_ in an identifier shall designate a character whose encoding in ISO 10646 falls into one of the ranges specified in Annex E. Upper- and lower-case letters are different. All characters are significant. *

[*Footnote: On systems in which linkers cannot accept extended characters, an encoding of the universal-character-name may be used in forming valid external identifiers. For example, some otherwise unused character or sequence of characters may be used to encode the "??u" in a universal-character-name. Extended characters may produce a long external identifier, but C++ does not place a translation limit on significant characters for external identifiers. In C++, upper and lower case letters are considered different for all identifiers, including external identifiers.]

Item 4) Augment the definition of _c-char_ in 2.10.2, Character Literals, as follows:

```
c-char:
   any member of the source character set except
       the single-quote ', backslash \, or new-line character
   escape-sequence
   universal-character-name
```

Then add a new paragraph 5, as follows:

5 A _universal-character-name_ is translated to the encoding, in the execution character set, of the character named. If there is no such encoding, the _universal-character-name_ is translated to an implementation-defined encoding. [Note: In translation phase 1 a _universal-character-name_ is introduced whenever an actual extended character is encountered in the source text. Therefore, all extended characters are described in terms of _universal-character-names_. However, the actual compiler implementation may use its own native character set, so long as the same results are obtained.]

Item 5) Augment the definition of _s-char_ in 2.10.4, String Literals, as follows:

```
s-char:
   any member of the source character set except
       the double-quote ", backslash \, or new-line character
   escape-sequence
   universal-character-name
```

Then, in paragraph 5, change "Escape sequences" to "Escape sequences and _universal-character-names". Change the last sentence to read as follows:

In a non-wide string literal, a _universal-character-name_ may map to more

than one char element. The size of a wide string literal is the total number of escape sequence, _universal-character-names_, and other characters, plus one for the terminating L'\0'. The size of a non-wide string literal is the total number of escape sequences and other characters, plus at least one for the multibyte encoding of each _universal-character-name_, plus one for the terminating '\0'.

Item 6) Add an annex to list the universal-character-names for identifiers.

Annex E (normative) Universal-character-names for Identifiers [extended-id]

---

1 This Clause lists the hexadecimal code values that are valid in _universal-character-names_ in C++ identifiers.

2 This table is reproduced unchanged from ISO/IEC PDTR 10176, produced by ISO/IEC JTC1/SC22/WG20, except that the ranges 0041-005a and 0061-007a designate the upper and lower case English alphabets, which are part of the basic source character set, and are not repeated in the table below.

[Editorial Note: If PDTR 10176 is changed during its balloting and adoption as a TR, then this table should be changed to match its changes.]

Latin: 00c0-00d6,00d8-00f6,00f8-01f5,01fa-0217,
0250-02a8,1e00-1e9a,1ea0-1ef9
Greek: 0384,0388-038a,038c,038e-03a1,03a3-03ce,03d0-03d6,03da,03dc,03de,
03e0,03e2-03f3,
1f00-1f15,1f18-1f1d,1f20-1f45,1f48-1f4d,1f50-1f57,1f59,1f5b,1f5d,
1f5f-1f7d,1f80-1fb4,1fb6-1fbc,1fc2-1fc4,1fc6-1fcc,1fd0-1fd3,
1fd6-1fdb,1fe0-1fec,1ff2-1ff4,1ff6-1ffc,
Cyrilic: 0401-040d,040f-044f,0451-045c,045e-0481,0490-04c4,04c7-04c8,
04cb-04cc,04d0-04eb,04ee-04f5,04f8-04f9
Armenian: 0531-0556,0561-0587
Hebrew: 05d0-05ea,05f0-05f4
Arabic: 0621-063a,0640-0652,0670-06b7,06ba-06be,06c0-06ce,06e5-06e7,
Devanagari: 0905-0939,0958-0962
Bengali: 0985-098c,098f-0990,0993-09a8,09aa-09b0,09b2,09b6-09b9,
09dc-09dd,09df-09e1,09f0-09f1
Gurmukhi: 0a05-0a0a,0a0f-0a10,0a13-0a28,0a2a-0a30,0a32-0a33,
0a35-0a36,0a38-0a39,0a59-0a5c,0a5e
Gujarati: 0a85-0a8b,0a8d,0a8f-0a91,0a93-0aa8,0aaa-0ab0,0ab2-0ab3,
0ab5-0ab9,0ae0,
Oriya: 0b05-0b0c,0b0f-0b10,0b13-0b28,0b2a-0b30,0b32-0b33,0b36-0b39,
0b5c-0b5d,0b5f-0b61,
Tamil: 0b85-0b8a,0b8e-0b90,0b92-0b95,0b99-0b9a,0b9c,0b9e-0b9f,0ba3-0ba4,
0ba8-0baa,0bae-0bb5,0bb7-0bb9,
Telugu: 0c05-0c0c,0c0e-0c10,0c12-0c28,0c2a-0c33,0c35-0c39,0c60-0c61,
Kannada: 0c85-0c8c,0c8e-0c90,0c92-0ca8,0caa-0cb3,0cb5-0cb9,0ce0-0ce1,
Malayalam: 0d05-0d0c,0d0e-0d10,0d12-0d28,0d2a-0d39,0d60-0d61,
Thai: 0e01-0e30,0e32-0e33,0e40-0e46,0e4f-0e5b,
Lao: 0e81-0e82,0e84,0e87,0e88,0e8a,0e0d,0e94-0e97,0e99-0e9f,0ea1-0ea3,
0ea5,0ea7,0eaa,0eab,0ead-0eb0,0eb2,0eb3,0ebd,0ec0-0ec4,0ec6,
Georgian: 10a0-10c5,10d0-10f6,
Hiragana: 3041-3094,309b-309e
Katakana: 30a1-30fe,
Bopmofo: 3105-312c,
Hangul: 1100-1159,1161-11a2,11a8-11f9
CJK Unified Ideographs: f900-fa2d,
fb1f-fb36,fb38-fb3c,fb3e,fb40-fb41,fb42-fb44,fb46-fbb1,fbd3-fd3f,
fd50-fd8f,fd92-fdc7,fdf0-fdfb,fe70-fe72,fe74,5e76-fefc,
ff21-ff3a,ff41-ff5a,ff66-ffbe,ffc2-ffc7,ffca-ffcf,ffd2-ffd7,
ffda-ffdc,4e00-9fa5