SC22/WG14 N799

Editor's Report

January 2, 1998

1. Status

As you know, an editorial committee consisting primarily of Benito, Farance, Jaeschke, Jones, MacDonald, and Walls (with the assistance of Thomas, Gwyn, and others) met in Santa Cruz November 17–20 to prepare the draft for CD ballot. After much work at the meeting and some subsequent formatting clean-up, a final document was prepared on November 24 and forwarded to SC22. A preliminary version of this draft was distributed (via the committee's web page) as N794; this document has been circulated outside the committee as well as within it and has generated a fair amount of discussion on the comp.std.c newsgroup as well as some private e-mail pointing out errors and omissions. The remainder of this report is based on these comments as well as my own review.

2. CD1 Errata

General

All of the headings are too small. When the size of the body text was increased, the sizes of the headings were not increased to match.

General

None of the tables are properly formatted in the text version due to a production error.

General

There are many typesetting issues still to be addressed. For example, there are many equations typeset as code and vice versa, particularly in annexes F and G. Some of these issues primarily affect the typeset version and some primarily affect the text version.

General

The term ‘‘universal-character-name’’ should not be hyphenated (other than in the syntax).

Clause 3, Paragraph 2, Page 3

The reference to ISO 2382-1 should be to ISO/IEC 2382-1.

5.1.1.2, Forward References, Page 10

There should be a forward reference to 5.2.1 for universal character names.

5.1.2.3, Example 6, Page 16

‘‘However’’ (4th line from the end) should be followed by a comma.

5.2.1.1, Footnote 12, page 19

The reference to ISO/IEC 646:1991 should be to ISO 646.

5.2.4.2.2, Paragraph 6, Page 26

The list of values should be indented like the one in paragraph 5.

6.1 and 6.2, Pages 31–66

The reorganization specified in N672 has not been applied.

6.1.2, Paragraph 2, Page 34

The reference to annex H should be to annex I.

6.1.2.5, Footnote 30, Page 40

In the last line, there should not be a comma after ‘‘two’’ and ‘‘it’’ should be ‘‘is’’.

6.1.2.6, Paragraph 1, Page 42

Near the end of the fifth line, the period after ‘‘following requirements’’ should be a colon.

6.1.4, Paragraph 6, Page 55

Although this paragraph doesn't contain the magic word ‘‘unspecified’’, whether string literals are distinct or not is unspecified and should be added to K.1.

6.1.5, Paragraph 1, Page 55

The definition of operator is missing **`{`** and **`}`**, although it does have the alternative spellings **`<:`** and **`:>`**. They should also appear in the constraint. (Also annex B.)

6.1.6, Paragraph 1, Page 56

The definition of punctuator is missing **`..`** (Also annex B.)

6.2.1.1, Paragraph 2, Page 60

The period after ‘‘may be used’’ should be a colon.

6.2.1.3, Footnote 48, Page 61

In the last line, there should be a space after the comma.

6.3.2.2, Paragraphs 5–6, Page 72

These paragraphs are almost impossible to read correctly. At the very least, paragraph 6 should be a continuation of paragraph 5, not a new paragraph. My suggestion is to keep paragraph 6 and move the end of paragraph 5 starting at the fifth line (‘‘If the function is defined...’’) to the end of paragraph 6.

6.3.2.5, Example 5, Page 78

In the last line of code, there should be a space before the square brackets.

6.3.3.3, Paragraphs 2–4, Page 81

In each paragraph, ‘‘the integer promotion is’’ should be ‘‘the integer promotions are’’.

6.3.6, Paragraph 12, Page 87

In line 2, ‘‘know constant size’’ should be ‘‘known constant size’’.

6.3.15, Paragraphs 4–6, Page 93

The last clause of paragraph 4 should read:

> the result of the operator is the value of the second or third operand (whichever is evaluated), converted to the type described below.

The first sentence of paragraph 5 should read:

> If both the second and third operands have arithmetic type, the type that the usual arithmetic conversions would yield if applied to those two operands is the type of the result.

The last sentence of paragraph 6 should read: ‘‘in which case the type of the result is pointer to **`void`**.’’

6.5.2, Paragraph 4, Page 103

In the middle of the second line, ‘‘specified’’ should be ‘‘specifier’’.

6.5.2.3, Paragraph 3, Page 108

In the second line, ‘‘complete’’ should be ‘‘incomplete’’.

6.6.4.1, Paragraph 3, Page 142

In the first line, ‘‘preceeding’’ should be ‘‘preceding’’. (And I’d be grateful if anyone could explain to me why ispell doesn’t complain about it.)

6.6.6.1, Example 2, Page 146

In the middle of the block of code, the line ‘‘**`goto lab 4;`**’’ should be ‘‘**`goto lab4;`**’’.

6.8.8, Paragraph 1, Page 170

It might be clearer to specify that **`__LINE__`** is the line number *within the current source file* and that **`__FILE__`** is the name of the *current* source file.

6.8.8, Paragraph 2, Page 171

The second line should not appear and the remainder of the paragraph should be formatted as definitions as in the previous paragraph.

6.8.9, Paragraph 1, Page 172

In the third line, the period after ‘‘follows’’ should be a colon.

6.8.9, Examples, Paragraph 2, Page 172

In the **`#pragma`**, ‘‘**`list`**’’ should be ‘‘**`listing`**’’.

7.1.2, Paragraph 1, Page 175

In the last line, ‘‘explicity’’ should be ‘‘explicitly’’.

7.1.7, Paragraph 2, Page 179

In the last line, ‘‘for’’ should be ‘‘of’’.

7.1.8, Paragraph 3, Page 181

‘‘return’’ should be ‘‘returns’’. (Also annex C.)

7.2.1.1, Paragraph 2, Page 183

In line 4, ‘‘name of the source file, and the source line number’’ should be ‘‘name of the source file, the source line number, and the name of the enclosing function’’.

7.3.1.3, Page 186

The isblank function was not approved by the committee and should not have appeared in the draft.

7.4, Paragraph 3, Page 190

Starting at the end of the first line, ‘‘character constants’’ should be ‘‘constants’’.

7.4.5, Paragraph 1, Page 199

The reference to footnote 151 should be to 149 instead.

7.5, Paragraph 3, Page 202

The reference to footnote 153 should be moved to the end of the sentence.

7.5.1, Paragraph 2, Page 203

In the very last line, there should be a reference to **`strfxtime`** as well as **`strftime`**.

7.7.8.3, Paragraph 3, Page 236

In the last line, ‘‘too large’’ should be ‘‘too large or too small’’.

7.11, Paragraph 3, Page 266

The portion of the paragraph between the two lists should read:

> which expand to constant expressions with distinct values that have type compatible with the second argument to, and the return value from, the **`signal`** function, and whose values compare unequal to the address of any declarable function; and the following, which expand to positive integer constant expressions with type **`int`** and distinct values that are the signal numbers...

7.13.6.1, Paragraph 3, Page 288

In the list item beginning ‘‘An optional **`hh`**...’’, the words for **`hhn`** have been omitted.

7.13.6.1, Paragraph 14, Page 293

The header should read "Environmental limits".

7.13.6.2, Paragraph 11, Pages 297–298

In the descriptions of the **`s`**, **`[`**, and **`c`** format specifiers, there should not be a paragraph break before the paragraph beginning ‘‘If an **`l`** qualifier is present...’’.

7.14.5, Footnote 227, Page 336

The last line of the footnote is missing. It should read:

```
(char *)p < (char *)base + nmemb * size
```

7.16.1, Paragraph 1, Page 357

In the first line, ‘‘four types and several functions’’ should be ‘‘several types and functions’’.

7.16.2.3, Paragraphs 5–6, Page 360

The last line of paragraph 5 and all of paragraph 6 should be deleted.

7.16.2.6, Paragraph 3, Page 363

At the end of the block of **`//`** comments, there should be one more:

> **`//`** *if the offset cannot be determined.*

7.16.3.6, Paragraph 6, Page369

‘‘**`%P`**’’ should be ‘‘**`%p`**’’ and "am" and "pm" should not be in program font.

7.18.2.1.1, Paragraph 2, Page 373

‘‘Th’’ should be ‘‘The’’.

7.18.2.1.3, Page 373

The iswblank function was not approved by the committee and should not have appeared in the draft.

7.18.2.2, Paragraph 1, Page 376

The reference to subclause 4.5.2.1 should be to 7.18.2.1.

7.19.2.1, Paragraph 2, Page 381

The last sentence is duplicated.

7.19.2.1, Paragraph 2, Page 382

In the list item beginning ‘‘An optional **`hh`**...’’, the words for **`hhn`** have been omitted.

7.19.2.1, Paragraph 14, Page 388

The header should read "Environmental limits".

7.19.2.1, Paragraph 16, Page 388

This should be a normal Forward References section — it should not have a paragraph number and the header should be in bold font.

7.19.2.2, Paragraph 11, Pages 391

In the second line of the description of the **[** format specifier, ‘‘f’’ should be ‘‘If’’.

7.19.2.2, Paragraph 11, Pages 391–392

In the descriptions of the **s**, **[**, and **c** format specifiers, there should not be a paragraph break before the paragraph beginning ‘‘If an **l** qualifier is present...’’.

7.19.2.2, Examples, Page 393

There should not be numbered paragraphs in the examples.

7.19.2.2, Paragraph 22, Page 394

This should be a normal Forward References section — it should not have a paragraph number and the header should be in bold font.

7.19.7.3.1, Paragraph 4, Page 426

This should be a normal Forward References section — it should not have a paragraph number and the header should be in bold font.

7.19.7.3.3, Paragraph 3, Page 426

‘‘a value between’’ should be ‘‘or a value between’’.

Annexes

The annexes and the index are all formatted wider than the main text.

Annex A, Page 434

The hyphens in the standards’ designations should be dashes.

F.9.3.11, Paragraph 1, Page 501

In the second bullet item, there should be a space between **)** and ‘‘returns’’.

G.5, Paragraph 5, Page 514

The reference to subclause 7.9.2 should be to 7.8.2.

Index, Pages 559–570

The index, which was produced manually, is both incomplete and inaccurate. I have since reviewed and corrected it and used that information to continue the process of adding index macros to the text so as to generate the index automatically as part of the formatting process. I have also added the automatic index generation to the formatting process and have produced an enhanced and corrected index to CD1 (N800) which can be used to aid in the review process.

Note that there is still a lot of work to be done on the index; I would appreciate any suggestions for additional entries, additional cross references, missing references, or excessive references.

3. Suggested Editorial Changes

General

All hyphenated terms should be examined to determine where the hyphenation is appropriate and where it isn't. All italicized terms should also be examined.

General

References to other standards should not include specific versions except in the References and Bibliography.

5.1.1.2, Paragraph 6, Page 9

Since we now allow concatenation of character and wide string literals, this paragraph should be rewritten to simply:

> Adjacent string literal tokens are concatenated.

5.2.1 Paragraph 4, Page 16

Universal character names should have their own subclause (like trigraphs do).

I also suggest changing the description to:

> The universal character name **\U***nnnnnnnn* designates the character whose character short identifier (as specified by ISO/IEC 10646-1) is *nnnnnnnn*. Similarly, the universal character name **\u***nnnn* designates the character whose character short identifier is 0000*nnnn*.

5.2.4.2.2, Paragraph 9, Page 28

A better wording is:

> The values given in the following list shall be replaced by implementation-defined expressions with [positive?] values that are less than or equal to those shown:

6.1.2.5, Footnote 29, Page 40

I suggest rewording the first two sentences to:

> An implementation may define new keywords that provide alternative ways to designate a basic (or any other) type; this does not violate the requirement that all basic types be different.

6.1.3.1, Paragraph 1, Page 47

The definition of hexadecimal digit really belongs in 6.1.3.2 (Integer constants) since there is text there explaining it. It would perhaps be best to switch these two sections.

6.1.3.4, Paragraph 5, Page 50

The lists of types for constants would be much easier to read as a table.

6.5.2.2, Paragraph 5, Page 108

It would be better to say:

> The type is incomplete until after the **}** that terminates the list.

since that's what we say for structure and union specifiers (cf., 6.5.2.1, Paragraph 6, Page 104).

Clause 7

Many of the subclauses should be rearranged to put them into alphabetical order.

7.4.5, Paragraph 1, Page 199

Since footnote 149 (which is what this is supposed to refer to) is so far away, it should be replicated here or this subclause should be moved to immediately after 7.4.2.

7.7, various

A number of subclauses in this section write out equations using words such as ''x raised to the power y''. I suggest using equations in all cases.

7.7.6.5, Paragraphs 2–3, Page 230

This subclause uses ‘‘base-ten’’, other subclauses use ‘‘base-2’’; we should be consistent. I suggest using numbers everywhere.

7.13.3, Paragraph 7, Page 279

Since this paragraph is talking about the standard streams, it might be better to move it to subclause 7.13.2 (Streams).

7.16.2.4, Paragraph 2, Page 361

At the end of the line, ‘‘takes into account of the additional members’’ should be ‘‘takes into account the values of the additional members’’.

7.18.2.1, Paragraph 2, Page 372

In the second line, it would be better to not mention the exact number of functions (it's easy to overlook if the list ever changes). Similarly for 7.18.2.2.1 paragraph 3 (page 376) and 7.18.2.2.2 paragraph 3 (page 377).

H.2.4, Paragraph 1, Page 525

The ugly arrows should be replaced by real arrows.

K.2, Page 537, Bullet 3

Similarly, converting between a pointer to an object or incomplete type and a pointer to a function causes undefined behavior. Subclause 6.2 is also relevant.

K.2, Page 537, Bullet 12

It should be clarified that this only applies to relational comparisons, not equality comparisons.

K.2, Page 538, Bullet 14

I think what this is supposed to say is ‘‘An attempt is made to access an object through both a restrict-qualified pointer and another pointer not based on it.’’

4. Suggested Possibly-substantive Changes

3.18, Paragraph 3, Page 6

An implementation need not successfully translate a given program if it exceeds a translation limit.

6.1.3.1, Paragraph 1, Page 47

In the first two productions for hexadecimal floating constant, the binary exponent part should be optional (parallel to decimal floating constant).

5. Open Issues

Clause 4, Paragraph 2, Page 7

Should **`<stdbool.h>`** be added to the list of required headers for freestanding implementations?

6.1, Paragraph 3, Page 31

Are all the italicized terms really appropriate?

6.1.3.1, Paragraph 1, Page 47

The **`0x`** and **`0X`** prefixes are used in multiple places, it might simplify the grammar to factor them out into their own nonterminal.

Clause 7

There are no synopses for the float and long double versions of the math routines and there are a number of synopses that basically duplicate or refer to other synopses. We should consider allowing multiple synopses for a family of functions with a single description like the Unix man pages have traditionally done.

7.7.11.2, Paragraph 2, Page 242

It is not clear what the last line (‘‘a call to the **`nan`** function is unspecified’’) is supposed to mean.

7.11.2.1, Paragraph 3, Page 268

It is not clear when the **`raise`** function is permitted to fail; is it only for an invalid signal number or are there other conditions?

7.13.5.2, Paragraph 4, Page 283

It is not clear what happens if **`stream`** is a null pointer and a write error occurs on one of the streams being flushed.

7.13.6.2, Paragraph 11, Page 298

In the description of the `[` conversion specifier, it is not clear whether the scanset is composed of single byte characters or multibyte characters.

7.14.3.4, Paragraph 3, Page 334

The comparison described in the last sentence results in undefined behavior if the object has moved.

7.16.1, Paragraph 4, Page 358

It is not always clear what happens if a member is outside its normal range when calling functions in the subclause.

7.18.2.1, Pages 372–377

Many of these functions provide very little guidance as to their intended semantics other than the similarity between their names and the names of the ordinary character functions. It seems like we should say a bit more than we currently do.

Annex F

There are no entries for **`ilogb`**, **`scalbln`**, **`llrint`**, **`llround`**, or **`nextafterx`**. We should say something about them, just so it doesn’t look like an oversight.

K.2, Page 535, Bullet 4

The first character of an identifier can’t be a digit, the syntax doesn’t allow it.

K.2, Page 540, Bullet 10

This isn’t undefined behavior. I think the situation that this is supposed to be describing is that when a macro expansion ends with a macro name which is ‘‘painted blue’’, and that macro is a function-like macro, and the first token of the remaining source-file tokens is a `(`, it was not clear whether that macro was to be replaced or not and thus it was undefined behavior to depend on one or the other.

Rereading the subclause on rescanning again, it now looks to me like it is clearly specified that the macro is not replaced. Does anyone disagree?