

Slay Some Earthly Demons VIII

Document: n3346

Author: Martin Uecker

Date: 2024-09-08

This remaining issues from n3244 were split up into the following series:

- n3340 - Slay Some Earthly Demons II
- n3341 - Slay Some Earthly Demons III
- n3342 - Slay Some Earthly Demons IV
- n3343 - Slay Some Earthly Demons V
- n3344 - Slay Some Earthly Demons VI
- n3345 - Slay Some Earthly Demons VII
- n3346 - Slay Some Earthly Demons VIII**
- n3347 - Slay Some Earthly Demons IX

Meeting meeting notes about previous discussion can be found in n3281. The following numbers correspond to N3220 and the ones with (→) to N3301.

- J.2.19 needs more work, cf. SC22WG14.25600 (**note to editor:** parts applied by mistake)
- J.2.21 done (incl. additional change), (**note to editor:** there is a “not“ missing in 6.3.2.1)
- n3340 J.2.40 (→ J.2.38) withdrawn in June, now revised (cf. SC22WG14.25600)
- J.2.56 done
- J.2.57 done (option 1)
- n3341 J.2.58 (→ J.2.54) not discussed in June
- J.2.60 (→ J.2.56) direction to editor (**note to editor:** not yet done)
- n3342 J.2.63 (→ J.2.59) no consensus in June, now revised to make implementation-defined
- n3343 J.2.63 wording change not related to an entry in Annex J
- J.2.67 done
- J.2.69 done (alternative wording for semantic section)
- n3344 J.2.75 (→ J.2.70) now revised (cf. SC22WG14.25600)
- J.2.77 (→ J.3.13. 2) not discussed in June, but fixed editorially
- J.2.78 (→ J.3.13. 3) not discussed in June, but fixed editorially
- n3345 J.2.79 (→ J.2.72) not discussed in June
- n3346 J.2.80-J.2.82 (→ J.2.73-J.2.75) not discussed in June (cf. SC22WG14.25600)**
- n3347 J.2.87 (→ J.2.80) not discussed in June

Acknowledgments: Thanks to Joseph Myers, David Svoboda, Chris Bazley, and the UB study group for helpful comments and corrections.

Annex J (J.2.73-75 N3301, J.2.80-82 N3220)

(73) The initializer for a scalar is neither a single expression, nor an empty initializer, nor a single expression enclosed in braces (6.7.11).

(74) The initializer for a structure or union object is neither an initializer list nor a single expression that has compatible structure or union type (6.7.11).

(75) The initializer for an aggregate or union, other than an array initialized by a string literal, is not a brace-enclosed list of initializers for its elements or members (6.7.11).

Analysis

It does not seem useful to have undefined behavior in initializers.

Update: The clause that the encoding prefixes have to match was added to §7 and the green marking of the first constraint was removed (as it already exists).

Proposed Change

Constraints

4 The type of the entity to be initialized shall be an array of unknown size or a complete object type. An entity of variable length array type shall not be initialized except by an empty initializer. An array of unknown size shall not be initialized by an empty initializer.

5 The initializer for a scalar shall be a single expression, optionally enclosed in braces, or it shall be an empty initializer.

6 The initializer for an object that has structure or union type shall be either a single expression that has compatible type or a brace-enclosed list of initializers for the elements or named members.

7 The initializer for an array shall be either a string literal, optionally enclosed in braces, or a brace-enclosed list of initializers for the elements. An array initialized by character string literal or UTF-8 string literal shall have a character type as element type. An array initialized with a wide string literal shall have element type compatible with a qualified or unqualified wchar_t, char16_t, or char32_t, and the string literal shall have the corresponding encoding prefix (L, u, or U, respectively).

Semantics

12 ~~The initializer for a scalar shall be a single expression, optionally enclosed in braces, or it shall be an empty initializer.~~ If the initializer **for a scalar** is not the empty initializer, the initial value of the object is that of the expression (after conversion); the same type constraints and conversions as for simple assignment apply, taking the type of the scalar to be the unqualified version of its declared type

14 ~~The initializer for a structure or union object shall be either an initializer list as described below, or a single expression that has compatible structure or union type. In the latter case, If~~ **the initializer for a struct or union object is a single expression,** the initial value of the object,

including unnamed members, is that of the expression.179)

15 **For** an array of character type ~~may be~~ initialized by a character string literal or UTF-8 string literal, ~~optionally enclosed in braces~~: successive bytes of the string literal (including the terminating null character if there is room or if the array is of unknown size) initialize the elements of the array.

16 **For** an array ~~with element type compatible with a qualified or unqualified wchar_t, char16_t, or char32_t may be~~ initialized by a wide string literal, ~~with the corresponding encoding prefix (L, u, or U, respectively), optionally enclosed in braces~~: Successive wide characters of the wide string literal (including the terminating null wide character if there is room or if the array is of unknown size) initialize the elements of the array.

~~17 Otherwise, the initializer for an object that has aggregate or union type shall be a brace-enclosed list of initializers for the elements or named members.~~