

The Kona Votes

on Revisions to ANSI/ISO C

Robert Jervis
 robert.jervis@sun.com
 (415)336-7964

Thursday night of the Kona X3J11/WG14 meeting in Kona Hawaii we held an ad hoc session on which revisions to C we were interested in making. About twenty of the committee members attended. We collected 57 ideas, which vary from specific and small to vague and far reaching. We then took a preference vote on each topic. Members could vote yes, no or don't care. The topics are listed below, sorted more or less by number of yes votes.

These votes should be taken as guidance. Topics with popular support will have a good chance of being accepted. People interested in seeing these topics added to C should therefore prepare papers proposing specific wording to implement the suggested change. Topics with high negatives, or low interest will have a very difficult time getting accepted, so writing a paper is probably not worth the effort.

Of course, none of these votes are final. Some popular ideas may not be adopted because of backward compatibility or limitations of some hardware implementation. Other popular ideas may prove difficult to make precise, or when made precise prove unpalatable. Dark horse topics may also be adopted if the right specific proposal is submitted.

Also, the fact that a proposal doesn't appear below doesn't mean that it can't be suggested later. At this stage any proposal championed at a meeting will get consideration.

I have tried to provide a brief description of each topic. I apologize to anyone whose topic I misrepresent. Some of the topics are vague and some were vague even to the ones proposing them.

(20 voting)
 yes-no

Majority favored

- | | |
|------|---|
| 18-0 | Incorporate Defect Reports |
| 18-0 | Longer external names (current requirement is 6 significant chars) |
| 17-3 | Enhanced C++ compatibility |
| 15-1 | Remove default int (static x; would become illegal, instead you would have to write static int x;) |
| 14-0 | Float and long double math libraries. Math.h reserves the name space, this would mandate implementation of the functions. |

- 14-1 Compound literals / designated initializers (an NCEG proposal that would allow structure constants in expressions and initialization of designated structure and array elements)
- 14-1 Extended integers (another NCEG proposal; this one proposes a header containing typedefs for specific integer sizes up to 64 bits)
- 14-2 `bool` type
- 14-3 Remove obsolescent features.
- 14-3 `//` comments
- 13-2 Strongly typed enums (from C++: enums are distinct types, not synonyms for integer types, integers and enums cannot be freely mixed)
- 12-0 Internationalized identifiers (kanji identifiers in Japan, etc.)
- 12-2 Type safe linkage (from C++)
- 12-3 Inline functions
- 12-3 Arrays as first class objects (array assignment, parallel arithmetic on array objects)
- 12-5 Declaration statements (from C++: declarations can appear anywhere in a block, and even in some statements like `for`)

At least one third favored

- 10-1 Library cleanup
- 10-5 Overloading (function and/or operator)
- 10-6 Complex arithmetic (from NCEG)
- 9-1 Character set improvements (simpler multibyte support library)
- 9-2 Float point extensions (from NCEG: IEEE bindings, new relational operators, generic math functions, IEEE environment controls, etc.)
- 9-3 Range types
- 9-4 Restrict pointers (from NCEG: restrict pointers allow for better optimization of C code)
- 9-4 Classes in C (from paper WG14/N298 X3J11/93-044, based on C++: single inheritance, virtual functions and operator `new`)
- 9-4 Allow trailing comma in enum list
- 9-5 Modules (like Modula-2, etc.)
- 9-6 Reference types (from C++)
- 8-1 Bit length ints (ability to declare integers of specific size in bits, other than just as a bit field)
- 8-1 Better `stdio.h` (use your imagination)
- 8-3 Variable length arrays (from NCEG: passing variable length array parameters as well as automatic VLAs)

- 8-3 Discontiguous / distributed objects
- 7-3 String type
- 7-4 Constructors / destructors

Less than one third favored

- 6-3 DPCE (from NCEG: Data Parallel C Extensions)
- 5-3 Zero sized objects
- 5-3 Parallel i/o
- 5-4 Support for LIA (language independant arithmetic), CLID (language independent data types?)
- 4-0 POSIX localedefs
- 4-1 Iterators (from NCEG)

Tied

- 6-6 X3H5 control parallelism
- 3-3 ISO 10646 support
- 3-3 Namespaces (from C++)

Disfavored, but by less than half

- 6-7 Statement functions (placing functions in expressions)
- 5-9 New user defined operators
- 4-6 Remove float, long double name reservations from math.h
- 4-6 Sets
- 3-6 Debugging support
- 3-8 Exceptions (from C++)
- 2-4 Automatic management of dynamic data structures
- 2-6 Fixed point arithmetic types
- 1-6 label type
- 1-9 Logarithmic arithmetic type
- 0-9 Garbage collection

Disfavored by more than half

3-11	Nested functions
1-11	Dynamic casts (from C++)
1-15	Templates (from C++)
0-17	Multiple inheritance (from C++)