Document Number: WG14 N609/X3J11 96-073

C9X Revision Proposal

Title: Boolean as a new type
Author: Frank Farance
Author Affiliation: Farance Inc.
Postal Address: 555 Main Street, New York, NY, 10044-0150, USA
E-mail Address: frank@farance.com
Telephone Number: +1 212 486 4700
Fax Number: +1 212 759 1605
Sponsor: X3J11
Date: 1996-09-13
Proposal Category:
__ Editorial change/non-normative contribution
__ Correction
X_ New feature
__ Addition to obsolescent feature list
__ Addition to Future Directions
__ Other (please specify) _____
Area of Standard Affected:
__ Environment
X_ Language
__ Preprocessor
__ Library
__ Macro/typedef/tag name
__ Function
__ Header
Prior Art: C++
Target Audience: C programmers
Related Documents (if any):
Proposal Attached: X_ Yes __ No, but what's your interest?

This proposal is for adding boolean as a new type, not just a macro for an existing integer type. The following problems need to be resolved with a boolean type:

1. It is a keyword, or hidden magic turned on by including a header?

2. What is its type? A new C type or an existing one?

3. What are the promotion rules?

4. What about use of "false" and "true" in the preprocessor?

5. What is the migration strategy?

Answer #1: For compatibility reasons, we need to have a header that turns on this feature: "#include <bool.h>". The type is "bool" with values "false" and "true".

Answer #2: In previous proposals, the type "bool" was a macro (or typedef) for an existing C integral type. This has raised some issues with respect to promotion rules for certain scenarios. I propose that we make "bool" a real, separate C type.

Answer #3: Promotion from "bool" to other types produces the values 0 or 1. Demotion from other types to bool follows the rules: equal to 0 ==> false, not equal to 0 ==> true.

Answer #4: The macros "false" and "true" will equate to 0 and 1 respectively.

Answer #5: The header "<bool.h>" will be used to avoid breaking existing programs. The macro

```
__bool_true_false_are_defined
```

will be defined when the implementation defines (as macros or keywords) the boolean type via the "#include <bool.h>" header. We should indicate that it is our intent to make "bool" a keyword in the next revision of the Standard.