From: Frank Farance
Organization: Farance Inc.
Telephone: +1 212 486 4700
Fax: +1 212 759 1605
E-mail: frank@farance.com
Date: 1997-06-26
Document Number: WG14/N738 J11/97-101
Subject: Improved boolean wording.

1. BOOLEAN KEYWORD SIMILAR TO COMPLEX

In subclause 6.1.1, Keywords, add the keyword "bool", "true", and "false". Append the following wording to the end of the Semantics section:

The tokens "bool", "true", and "false" are keywords in translation units where the header "<bool.h>" are included, but not otherwise; if the header is included, then any use of the tokens "bool", "true", and "false" prior to the first inclusion results in undefined behavior.

2. ADDING BOOL AS A BASIC TYPE

In subclause 6.1.2.5, types, add after paragraph #1 the paragraph:

> An object declared as the type "bool" is large enough to store the values "true" and "false" and shall have values 1 and 0, respectively.

In subclause 6.1.2.5, Types, change the first sentence in paragraph #10 from:

> The type "char", ...

to

> The type "bool", the type "char", ...

3. ADDING BOOL AS AN INTEGRAL TYPE

In subclause 6.1.2.5, Types, change the first sentence in paragraph #16 from:

> The type "char", ...

to

> The type "bool", the type "char", ...

4. PROMOTION TO INT

In subclause 6.2.1.1, Characters and integers, change the title to:

6.2.1.1 Boolean, characters, and integers

In subclaused 6.2.1.1, paragraph #1, change:

A "char", ...

to:

A "bool", a "char", ...

5. DEMOTION TO BOOL

In subclause 6.2.1.1, Characters and integers, add new paragraph #3:

When a value is demoted to a "bool", the result is "false" if the value equals 0, and "true" otherwise.

6. UNRESOLVED ISSUES

6.1 Extensions to printf/scanf

These have yet to be resolved in either the previous Boolean proposal (currently approved for addition to the C9X draft), or in the current proposal. A separate paper should be written.

6.2 Making logical and relational operators Boolean

The result of these expressions is either 0 or 1. Since there is no overloading in C (unlike C++), there is no need for changing the resultant type of these operators.

6.3 Extending the "is*" macros to return Boolean

This is a good idea, but a separate issue from either the previous or current Boolean proposal. A separate paper should be written.

6.4 Boolean bit fields

The goal is to have Boolean values, stored as a single bit in a structure. For example:

```
bool x:8
```

However, what happens with the following?

```
bool x:8
```

Getting the correct wording would be difficult, especially considering the DRs in this area. A solution that is workable on all systems is:

```
unsigned int x:1
```

```
        x = (bool) value
```

A paper could be written to allow this, but the wording would be tricky.