Document Number: X3J16/91-0093
WG21/N0014
Date: July 3, 1991
Project: 738-D
Reply to: Dan Saks
dsaks@wittenberg.edu

WG21 Meeting No. 1
X3J16 Meeting No. 6
Joint Session
June 19, 1991

Grand Hotel
Bantorget 1
Lund, Sweden

Lenkov convened the joint session of WG21 and X3J16 at 1:45 pm.

Gautron presented a proposal to allow 8-bit characters in C++ identifiers (WG21/N0003 = 91-0070).

Ball said we should define what a letter is rather than just say any 8-bit character can be used in an identifier.

Koenig explained that C has loose constraints on the character set. C enumerates the characters that a conforming implementations must support. C also says that digits must use contiguous codes in ascending order. It says almost nothing else. There are machines with 7-bit characters. Those machines can implement C. Do we want to prevent them from implementing C++?

Simonsen said that Gautron's proposal is covered by a more complete proposal under development by WG14. The WG14 proposal also includes widechars (like Kanji) not covered by Gautron's proposal.

Gautron replied that the WG14 proposal might take a long time to implement. We can have Gautron's proposal for free, and have it now. Gautron said his proposal is not intended to solve all such problems for all character sets.

Plum said this is a problem that SC22 has asked all language standards committees to address.

After a little more discussion, the committee referred the issue to the Extensions WG.

Simonsen presented N0009 = 91-0086. This is the proposal approved by WG14 in Tokyo.

Stroustrup made the following points:

1. The proposed notation is different from what X3J16 agreed to at its last meeting. Overall Simonsen's proposal is an improvement.

2. The additional keywords should not be optional. They should be available at all times, not just when iso646.h is included.
3. Having keywords as defines lets people play preprocessor games. We shouldn't force people to use the preprocessor to get to the basic facilities of the language.

He proposed that C++ switch to using WG14's set of digraphs and WG14's macro names as keywords. He also suggested that X3J16 ask WG14 to reserve the identifiers just as they reserve extern library names.

Charney saw a problem with the digraph <: because it forces C++ programmers to use a space to separate < from :: . Otherwise <:: will be tokenized as <: followed by : . He recommended using a different digraph like </ . Simonsen said his my analysis showed that only three digraphs are available using < , namely, <% , <: and <, . He doesn't like <, because of its use a sequence operator. Stroustrup added he doesn't mind putting in a space now and then (between tokens) to solve this problem.

Koenig presented an example demonstrating the problem with implementing the new words as macros. Suppose header lib.h contains

```
// lib.h
...
    {
    int and;
    ...
    };
```

An application that writes

```
#include <iso646.h>
#include <lib.h>
```

won't work. Therefore, the author of a header file must assume that *iso646.h* has been included, and must treat all *iso646.h* identifiers as reserved words in all cases. Now, suppose the author of lib.h wants to use these keywords as operators. Then the author must include *iso646.h* in case the user didn't, as follows

```
#include <iso646.h>
...
if (x < y and y < z) { ... }
```

Then any user who includes *lib.h* gets *iso646.h* turned on. That may break user code. Koenig concluded that using macros to implement these keywords is even more inconvenient than having them as keywords all the time.

Gautron said that an earlier proposal from AFNOR (via e-mail) suggested adding keywords for left and right shift. He suggested voting against that proposal.

Plum noted that C++ already has 16 new keywords not in C. Some of the proposed new keywords are needed, but others, like *bitand*, are added only for symmetry. Simonsen explained that keywords providing alternatives to !, like *not*, were added to cope with EBCDIC.

Straw vote: Who feels there are major technical problems with Simonsen's proposal (using keywords)? 0

Straw vote: Who thinks there are usability problems with the proposal? 8

Allison said this is an issue for DECUS users who are interested in moving from C to C++. There will be complaints from users about name space pollution, but he can't see a better way around the problem.

Stroustrup said that people writing portable programs are already avoiding and, or, and not.

Miller noted that using names beginning with two underscores or an underscore followed by an uppercase letter would avoid stealing names from the user name space.

Straw vote: Who thinks Stroustrup's proposal is complete and ready for vote? 16 yes, 3 no, 7 abstain.

Stroustrup said that using names beginning with two underscores or and underscore and an uppercase letter will interfere with implementors instead of users. Some suggested using *NotEq* instead of *not_eq*, but the names have already been chosen so that the most common names have the shortest spelling. We would also need an empty *iso646.h* for C and C++ compatibility. C++ cannot do with fewer keywords to implement this extension because C needs macro names to fake the keywords. For example, C++ cannot use *or=* because C can't write a macro with that name.

Miller suggested removing the *and* words from set to avoid stealing three words from user space.

O'Riordan suggested using <> for !=. Insinga advised against deviating from WG14. Simonsen agreed to take this suggestion back to WG14.

Straw vote: Who favors this proposal (to use digraphs and new words as keywords and to ask WG14 to reserve the keywords in C as with extern library identifiers)? 21 yes, 0 no, 5 abstain.

Lenkov closed the committee of the whole.

Lenkov adjourned the joint meeting of WG21 and X3J16 at 3:50 pm.