

C9X Revision Proposal

=====

Title: Removal of "auto" keyword

Author: David R. Tribble

Author Affiliation: Self

Postal Address: 6004 Cave River Dr.

Plano, TX 75093-6951

USA

E-mail Address: drt@wcwcen.wf.com

Telephone Number: +1 214 9641720 night

+1 214 9608649 day

Fax Number:

Sponsor:

Date: 1995-03-19

Proposal Category:

☐ Editorial change/non-normative contribution

☐ Correction

☐ New feature

☒ Addition to obsolescent feature list

☐ Addition to Future Directions

☐ Other (please specify) \_\_\_\_\_

Area of Standard Affected:

☐ Environment

☒ Language

☐ Preprocessor

☐ Library

☐ Macro/typedef/tag name

☐ Function

☐ Header

☐ Other (please specify) \_\_\_\_\_

Prior Art: None, although the "entry" keyword was removed in the previous ANSI/ISO C spec.

Target Audience: C programmers, and programmers who write programs that generate C code.

Related Documents (if any): None.

Proposal Attached: ☒ Yes ☐ No, but what's your interest?

Abstract: The "auto" keyword is rarely, if ever, used. It is redundant when it is used (to declare variables local to a function). Therefore, it should be deemed OBSOLETE as a reserved keyword.

===== Cover sheet ends here =====

PROPOSAL

The "auto" keyword shall be deemed OBSOLETE.

RATIONALE

The "auto" keyword is redundant and unnecessary. It is only used to declare variables with block scope within function bodies and compound statements (blocks) [?].

Most programmers never use the "auto" keyword, since it is implicit when defining local variables.

Some program generators might be affected, however, by the removal of the "auto" keyword.

## SYNTAX

This affects the lexical analysis of programs, because there is one less keyword.

## AFFECT TO EXISTING PROGRAMS

If a given implementation has removed the "auto" keyword from its lexicon, old programs may still be compiled successfully by adding this directive to the code:

```
#define auto
```

Note that the same effect can be gotten with most compilers by using a compiler option:

```
cc -Dauto= foo.c
```

## COMPATIBILITY

The "auto" may also be considered for obsolescence or removal in C++.

## IMPACT

This affects [6.1.1], [6.5.1], [6.7], [6.9.3]. These are mostly removals of references of "auto" variables.

This may (but probably doesn't) affect [6.1.2.2], [6.1.2.4], [6.5], [6.6.2], [6.7.1].