N2686: `#warning`

Javier A. Múgica - javier@aerotri.es

March 28th, 2021

Rationale

C has had since C89 an #error preprocessing directive. Over the years many implementations have provided a #warning directive, with its obvious semantics. Making this feature standard seems desirable and its implementation is very easy. It is currently supported by gcc, clang, EDG, the ICC and IBM's XL C at a minimum. Other compilers, like MSVC, support it via a #pragma.

Proposed wording

Include it in the list of directives in 6.10:

6.10 Preprocessing directives

Syntax

> **#** *error* *pp-tokens*$_{opt}$ *new-line*
> **#** *warning* *pp-tokens*$_{opt}$ *new-line*

Include it in section 6.10.5 Error directive, which would change its title to "Diagnostic directives".

6.10.5. ~~Error directive~~Diagnostic directives

Semantics

A preprocessing directive of ~~the~~either form

> **#** *error* *pp-tokens*$_{opt}$ *new-line*
> **#** *warning* *pp-tokens*$_{opt}$ *new-line*

causes the implementation to produce a diagnostic message that includes the specified sequence of preprocessing tokens.

Rationale for the wording

Many a reader will miss an extra sentence like:

> In addition, the **error** directive causes the translation to fail, as described in 4.

However, such a sentence is already missing in the current standard, and the C Rationale says:

> Formally, the Standard can require only that a diagnostic be issued when the `#error` directive is processed.