Support for #warning

Summary of Changes

R0

- Initial proposal

Introduction and Rationale

Almost all major C++ compilers support the `#warning` preprocessing directive to generate a diagnostic message from the preprocessor without stopping translation, as `#error` does, which can be useful for code authors who want to warn consumers of the code about non-fatal concerns. Consider a hypothetical library that does not yet support multithreading. The library author may wish to warn users about the lack of thread support but may not want to stop translation as the library can be safely consumed by an application that is also not multithreaded.

WG14 considered a similar proposal as part of WG14 N2686 at our Sept 2021 meeting and adopted the feature into C23 (straw poll results were: 17 in favor, 0 oppose, 1 abstain). The WG21 proposal is functionally identical to the WG14 proposal, with the only difference being due to existing variance in specification around how `#error` causes translation to stop. Adopting this proposal into C++ will be compatible with the same feature in C, and implementers appreciate keeping the preprocessor functionality the same between these two languages whenever possible.

Proposed Wording

The wording proposed is a diff from the working draft of WG21 N4892. Green text is new text, while ~~red~~ text is deleted text.

Modify the 15.1 [cpp.pre] grammar productions:

> control-line:
>
> `# include` *pp-tokens new-line*
> *pp-import*
> `# define` *identifier replacement-list new-line*
> `# define` *identifier lparen identifier-list$_{opt}$) replacement-list new-line*
> `# define` *identifier lparen* `...` *) replacement-list new-line*
> `# define` *identifier lparen identifier-list* `,` `...` *) replacement-list new-line*
> `# undef` *identifier new-line*
> `# line` *pp-tokens new-line*
> `# error` *pp-tokens$_{opt}$ new-line*
> `# warning` *pp-tokens$_{opt}$ new-line*

> # pragma *pp-tokens*_{opt} *new-line*
> # *new-line*

Modify the 15.8 [cpp.error] heading:

> 15.8 ~~Error directive~~Diagnostic directives

Modify 15.8p1:

> 1 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~~, and~~; the # error directive renders the program ill-formed.

Acknowledgements

I would like to recognize the following people for their help in this work: Jens Maurer, Javier Múgica, and Richard Smith.

References

[WG14 N2686]
#warning. Javier Múgica. http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2686.pdf