

# **The New C Standard** (Excerpted material)

---

**An Economic and Cultural Commentary**

**Derek M. Jones**

derek@knosof.co.uk

## 6.10.5 Error directive

### Semantics

`#error` A preprocessing directive of the form

```
# error pp-tokensopt new-line
```

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

### Commentary

Rationale The `#error` directive was introduced in C89 to provide an explicit mechanism for forcing translation to fail under certain conditions. Formally, the Standard can require only that a diagnostic be issued when the `#error` directive is processed. It is the intent of the Committee, however, that translation cease immediately upon encountering this directive if this is feasible in the implementation. Further diagnostics on text beyond the directive are apt to be of little value.

### C++

16.5p1 *... , and renders the program ill-formed.*

Both language standards require that a diagnostic be issued. But the C Standard does not specify that the construct alters the conformance status of the translation unit. However, given that the occurrence of this directive causes translation to terminate, this is a moot point.

### Common Implementations

Most implementation issue the diagnostic and then stop translation. However, a few implementations do continue translating after this directive is encountered. Some implementations also support the preprocessing directives `#inform`<sup>[1]</sup> and `#warning`<sup>[2]</sup>. As their identifier name suggests they generate informational messages and warnings respectively. The translation does not fail.

### Example

```
1  #include <limits.h>
2
3  #if CHAR_BIT != 9
4  #error This program only runs on a host where CHAR_BIT == 9
5  #endif
```

## References

1. Diab Data. *D-CC & D-C++ Compiler Suites User's Guide*. Diab Data, Inc, [www.ddi.com](http://www.ddi.com), 4.3 edition, June 1999.
2. R. M. Stallman. *Using the GNU Compiler Collection*. Free Software Foundation, Mar. 2004.