
Form-Feed and Vertical-Tab in // Comments

Doc No: **X3J16/93-0047**
WG21/N0254
c++std-env-368
Date: **October 21, 1993**
Project: Programming Language C++
Reply-To: Neal M Gafter
gafter@mri.com

1 Introduction

- (1) A preprocessing directive ends on a new-line token. On some systems, a form-feed or a vertical-tab may be treated as an end-of-line indicator, and therefore cause a new-line token to be introduced, while on others it may not. To prevent programs from depending on this system-specific behavior, form-feed and vertical-tab are not allowed between the tokens of a preprocessing directive. However, these whitespace characters may appear between the last token of a preprocessing directive and the end-of-line.
- (2) This problem was re-introduced into the C++ language with the line-ending comment. The same criteria does not suffice, because the body of the comment is not composed of preprocessing tokens, but rather it is composed of characters. An ideal solution is analogous to the ANSI/ISO C solution for preprocessing directives.

2 Proposal

- (1) I propose that we modify the Working Paper as follows. Change
- (2) The characters // start a comment, which terminates at the end of the line on which they occur.
- (3) to
- (4) The characters // start a comment, which terminates at the next end-of-line character. If there is a form-feed or vertical-tab character in such a comment, only white-space characters may appear between it and the end-of-line that terminates the comment.

3 Additional Observations

- (1) Sean Corfield pointed out the following vagueness in the current draft (and in ANSI/ISO C): Although form-feed and vertical-tab are explicitly disallowed between the tokens of a preprocessing directive, translation phase 3 allows an implementation to replace any non-empty sequence of white space characters (including a single form-feed or vertical-tab) with a single space. This effectively gives an implementation permission to avoid diagnosing a form-feed or vertical-tab that appears within a preprocessing directive. This proposal suffers from the same problem for // comments. I suggest we resolve this by defining both of these as *undefined* behaviors.