Use of the Escape Sequence ‘\A’

The character sequence "\AE\AE" is used in the code of the Tokeniser, Compactor and Parser.

The reason is that the logical design of the Tokeniser calls for its output to be a sequence of word tokens that include an initial and a terminating word token as sequence delimiters. The terminating word token was specifically designed to include the text “®®”, because ‘®’ suggests the ‘Return’ key and hence the end of a statement. (The ASCII character ‘®’ is now considered to represent a registered trademark).

Consequently the text “®®” is expected by the Compactor and passed through to the Parser.

‘®’ is in the extended part of the ASCII character set. It is not part of the basic ASCII character set expressible in 128 bits. ‘®’ is expressed as the decimal value ‘174’ and hex value ‘AE’. Hence in implementing the text “®®”, the text "\AE\AE" appears in the program code.

When the g++ compiler is used with the '-Wall' flag, "\AE\AE" results in the warning ***`unknown escape sequence:'\A'`***. It does not prevent the code compiling correctly (as far as is known). The text “\AE\AE” is not an error, as might be supposed from the compiler warning.

‘®’ could be replaced by the basic ASCII character expressed as the decimal value ‘13’ and hex value ‘0D’, and used to represent ‘CR’ or ‘enter/carriage return’.

However the change would have to be applied throughout the Tokeniser, Compactor and Parser, or a system error would be created.