

AdobeSM Customer Services

Calculating Characters per Pica on the Macintosh

This technical note describes how an approximate Characters per Pica (cpp) measurement can be calculated for any standard Adobe PostScript™ font.

BACKGROUND

Characters per Pica (cpp) measurements were used in standardizing the description of space requirements for specific fonts when laying out a page of printed material. The development and first implementation of cpp measurements began in the days of traditional typesetting, long before the desktop publishing era. As desktop computer technology matured, similarities in layout and type tools resurrected many of the concepts and terms developed by early typographers. These terms were transplanted into the digital arena, but lost much of their original meaning and value.

CPP – APPLICATION, DRIVER AND FONT

Characters per Pica is one example of a traditional term applied to digital typography to describe font characteristics. Its value is almost negated, however, by the effects of computer applications, drivers, fonts, and output devices. Each part of the desktop publishing puzzle adds variables to the final product that were not relevant when cpp was a standard.

On the Macintosh, font spacing is governed by three major variables:

- The application used
- The spacing information in the bitmap file for the font
- The printer driver

As a test, if you were to print the same three paragraphs of text from three different applications, you would most likely get three different cpp values for the same font, each one dependent on the application used to output — regardless of the font's original specification or “actual” cpp value.

CALCULATING APPROXIMATE CPP VALUES

Despite the discount to the meaning of cpp values on the Macintosh, they can still be useful in special cases. For these purposes, approximate cpp values can be determined as follows:

- 1 Determine the average width of a typeface design by using actual samples of text. Add the widths of all lowercase characters in a typeface's alphabet, a through z. This can be based on actual output or by adding values in the AFM files included with your font package. Call the result a value "T."
- 2 Determine a point size for which you would like to calculate a value. Typical sizes are 10 and 12 points. Call the size a value "p."
- 3 Multiply step 1 by step 2. Let that number be a value Tp.
- 4 To arrive at approximate cpp, plug Tp into the following equation:
$$340,722 \div Tp = \text{cpp}$$

The number 340,722 represents the result of algebraic equivalents accounting for the following:

- In PostScript, a single point is equivalent to 99.62% of a traditional typographer's point size.
- The sum of the widths of all lowercase characters from a to z must be divided by 26.5 to account for punctuation variances in real cpp output.
- Since a pica, by definition, is 12 points, the cpp value must be expressed in units of 12. Therefore, the total width of all lowercase characters, divided by 28.5 and multiplied by the point size, is normalized to units of 12.

In detail, the process is:

$$12 \div [(T \div 28.5) \times P] \div 1000]$$

See the descriptions of variables above.