

Why does Adobe PDF Library use five digits of precision?

The five digits of precision alludes to an ASFixed value. An ASFixed value is a hybrid of an Integer and a Real.

Where:

- An integer is a binary number with bits encoding values from 0 to 2^{32} (unsigned) or -2^{30} to 2^{30} (signed)
- A real / floating point splits the bits between representing significant figures and a mantissa for the exponent

An ASFixed value shifts the meaning of the bits of a signed 32 bit integer value from representing 2^{30} down to 2^0 , to representing values between 2^{15} (32,767) to 2^0 .

This leaves 15 bits to represent values 2^0 to 2^{-15} . 2^{-1} being 0.5 (1/2), 2^{-2} being 0.25 (1/4), etc., allowing for smaller rational numbers in between integers, but the precision won't be smaller than $1/(2^{15})$. If you take the log of 2^{15} , you get approximately 4.515.

For any number in a PDF that gets converted from a string to an ASFixed value, APDFL will read the full string. Given identical numbers before the decimal point, changes in the first four significant digits will result in the number being mapped to a different ASFixed value. If you also assume the first four significant digits are identical, then changes to the fifth digit **can** result in the number being mapped to a different ASFixed value, but that's not guaranteed. Anything beyond 5 digits won't affect the number's representation as an ASFixed value.

The number is only guaranteed to be mapped to different ASFixed values for the first four significant digits. This is because the full five digits for the value are not significant; rather, **almost** five digits are significant.

This means that numbers which are identical out to four significant digits, but have different fifth digit values, **could** be mapped to the same ASFixed value.

For example, xx.xxxx4 and xx.xxxx5 could potentially be mapped to the same ASFixed value.