Apple II
Technical Notes

Developer Technical Support

Apple IIGS
#79: Integer Math Data Types

Revised by: Jim Luther May 1990
Written by: Dan Strnad March 1990

This Technical Note describes the format of `Fixed` and `Frac` data types used by the Integer Math tool set and operations performed on the Integer Math numerical data types.
Revised since March 1990: Fixed original date, bit numbering of diagrams, and a multiplication sign in the equation.

As stated in Volume 1 of the *Apple IIGS Technical Reference*, the Integer Math tool set provides the following numerical data types:

```
Integers
Longints
Fixed
Frac
Extended
```

The precise format of the `Fixed` and `Frac` data types is not provided in the reference manual, so this Note details these formats.

The format for the `Fixed` data type is stated in the manual as being a 32-bit signed value with 16 bits of fraction. This means that the low-order 16 bits of the `Fixed` format data value are considered as a fraction of 2^16, which is the binary number represented by a one followed by 16 zeroes ($10000). In other words, a `Fixed` value is the same as a long integer value whose binary point has been moved to the left 16 places. In this representation, if the low-order part of the `Fixed` format data value were $8000, the fractional value would be equal to 1/2. A low-order part of $C000 would represent a fractional part equal to 3/4. Therefore the highest value that a `Fixed` can contain is 32,767 and 65,535/65,536; the least value is equal to -32768.

31	30	29		18	17	16
-32768	16384	8192	. . .	4	2	1

high-order word

15	14	13		2	1	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	. . .	$\frac{1}{16384}$	$\frac{1}{32768}$	$\frac{1}{65536}$

low-order word

Figure 1–Fixed Data Type

The format for the `Frac` data type is stated in the manual as being a 32-bit signed value with 30 bits of fraction. This means that the low-order 30 bits of the `Frac` format data value are considered as a fraction of 2^30, which is the binary number represented by a one followed by 30 zeroes ($40000000). In other words, a `Frac` value is the same as a long integer value whose binary point has been moved to the left 30 places. The high-order 2 bits of the `Frac` format data value are treated as follows. The high bit has a value of -2 and the low bit has a value of 1. Therefore the highest value that a `Frac` can contain is 1 and ((2^30)-1)/2^30; the least value is equal to -2.

31	30	29		18	17	16
-2	1	$\frac{1}{2}$	. . .	$\frac{1}{4096}$	$\frac{1}{8192}$	$\frac{1}{16384}$

high-order word

15	14	13		2	1	0
$\frac{1}{32768}$	$\frac{1}{65536}$	$\frac{1}{131072}$	. . .	$\frac{1}{268435456}$	$\frac{1}{536870912}$	$\frac{1}{1073741824}$

low-order word

Figure 2–Frac Data Type

Note that for `Longints`, `Fixed`, and `Frac` values, the hex representations of the largest and smallest data values are $7FFFFFFF and $80000000, respectively.

A property of the `Fixed` and `Frac` data types is that two `Fixed` or two `Frac` values may be added or subtracted just as if they were 32-bit integers. To demonstrate this, imagine scaling the numbers by a given factor to make them integers. After adding the numbers, the sum could be scaled back down by the same factor. This follows from the distributive property of multiplication over addition, which allows one to make the inference shown in the equations which follow. In these equations, `V1` and `V2` are both either `Fixed` or `Frac` values. The value for `C` being discussed, which illustrates the ability to scale `Fixed` and `Frac` values, is 2^16 for `Fixed` values of `V1` and `V2`, or 2^30 for `Frac` values of `V1` and `V2`.

$$\frac{(C * V1) + (C * V2)}{C} = \frac{C * (V1 + V2)}{C} = V1 + V2$$

Similarly, two Fixed or two Frac values may be compared, as Longints are compared, with one another. In general, the comparison, addition, and subtraction operations used for long integers may also be performed on any two Fixed or any two Frac values.

Further Reference

- *Apple IIGS Technical Reference Manual*
- *Apple Numerics Manual, Second Edition*