32-bit Data Types Integral data type

#include <Types.h>

long	signed long word	range: -1,247,483,648...1,247,483,647
unsigned long	unsigned long word	range: 0...4,294,967,295
Size	unsigned long word	range: 0...4,294,967,295
OSType	unsigned long word	range: 0...4,294,967,295. File creator,
OsType		file type. Usually four text chars such as... 'TEXT' or 'Guid' or 'WDBF' etc.
	typedef OSType ***OSTypePtr**;	
ResType		Also 4 chars. See **Standard Resource Types**
	typedef ResType ***ResTypePtr**;	
Fixed	signed long word	range: -32768 ... 32767.*xxxx*
		xxxx is 1-(1/65536)
	typedef Fixed ***FixedPtr**;	
Fract	signed long word	range: -2 ... 1.*yyyy*
		yyyy is 1-(1/1247483647)
	typedef Fract ***FractPtr**;	
float	floating-point	range: library-dependent

Notes: As a "Little Endian", the 68xxx CPU represents numbers in memory with the high-order bytes lowest in memory (contrasted to "Big Endians," such as the 80x86 which keeps the lowest-order byte of all data types at the lowest address). Take care to avoid reading 16-bit or 32-bit data which starts at an odd address; see 16-Bit Data Types for details.

The Size data type is used only in Memory Manager functions relating to Zone manipulation.

OSType and ResType values are used in File Manager and Resource Manager calls respectively. The convention for these data types is to use 4-byte constants containing readable parts of the ASCII character set. For displaying the value of one of these types, treat it as a 4-character string (lowest byte in memory is the first character, etc.):

```
OSType theFileType;                    /* assume we got via GetFInfo */
Byte       *bp = (Byte *)&theFileType;

printf( "The file type is '%c%c%c%c' \n",bp[0], bp[1], bp[2], bp[3]);
```

Fractional Types

Fixed data types are used in calls to **FontMetrics** , **SpaceExtra**, **SlopeFromAngle** and **AngleFromSlope** (along with the variety of math operations and conversions provided in the Toolbox Utilities).

A Fixed value has an implied 'binary point' between bit 16 and bit 15. The high-order word is the (signed) integer portion and the low-order word is the fractional portion. For instance, (Fixed)0x00010001 equates to 1 and 1/65536th (ie, approximately 1.000015258789).

The Fract data type is supported by the 128K ROMs. It allows very accurate representation of numbers between -2 and 2 (errors no worse than 1 part in one billion). Only the math and conversion functions of Toolbox Utilities use this data type directly, but it is handy in high-resolution graphics work (especially typesetting).

A Fract value has an implied 'binary point' between bits 30 and 29. Bit 31 is the sign bit, bit 30 is the one's place, and bits 29 through 0 constitute the fractional part. For instance, (Fract)0x40000001 equates to 1 and 1/1073741824.

The float data type is 32-bits long. Its implementation is compiler- and library-dependent.