

Cheryl Ewy
Steven Glass
Kenton Hanson

Integer Math Tools

August 7, 1986

Revision History

March 4, 1986	V00:00	Initial Release
April 22, 1986	V00:10	Int2Dec, Long2Dec, Dec2Int and Dec2Long calls modified
May 9, 1986	V00:20	Errors in the input/output lists for the math routines fixed
August 7, 1986	V00:30	Functions \$10 to \$21 expanded.

STANDARD TOOL SET CALLS

IMBootInit **Function number = \$01**

This call does nothing.

IMStartUp **Function number = \$02**

This call does nothing.

IMShutDown **Function number = \$03**

This call does nothing.

IMVersion **Function number = \$04**

Input	Word	Space for Result
Output	Word	Result

This call returns the version number for the Integer Math tool set.

IMReset **Function number = \$05**

IMReset is called when a system reset occurs. It does nothing.

IMActive **Function number = \$06**

Input	Word	Space for Result
Output	Word	Result

This call returns a non-zero result indicating that the tool set is active.

MATH ROUTINES

These routines come from the Macintosh and are used throughout the tool box. Several types of numbers are supported -

Integer	The common single word signed integer
Long Integer	The common double word signed integer
Fixed	A two word signed value with 16 bits of fraction
Frac	A two word signed value with 30 bits of fraction

Multiply

Function number = \$09

Input	LongWord	Space for Result
Input	Word	M1
Input	Word	M2
Output	LongWord	Result

Takes the two 16 bit inputs, multiplies them together and produces a 32 bit result . If the inputs were unsigned, the 32 bit result is unsigned. If the inputs were signed, the low word of the 32 bit result is the signed result.

SDivide

Function number = \$0A

Input	Word	Space for Remainder
Input	Word	Space for Quotient
Input	Word	Numerator
Input	Word	Denominator
Output	Word	Remainder
Output	Word	Quotient

Takes the two 16 bit signed inputs and divides them producing two 16 bit signed results.

UDivide

Function number = \$0B

Input	Word	Space for Remainder
Input	Word	Space for Quotient
Input	Word	Numerator
Input	Word	Denominator
Output	Word	Remainder
Output	Word	Quotient

Takes the two 16 bit unsigned inputs and divides them producing two 16 bit unsigned results.

LongMul

Function number = \$0C

Input	LongWord	Space for Result
Input	LongWord	Space for Result
Input	LongWord	M1
Input	LongWord	M2
Output	LongWord	Result (most significant)
Output	LongWord	Result (least significant)

Takes the two 32 bit inputs, multiplies them together and produces a 64 bit result . If the inputs were unsigned, the 64 bit result is unsigned. If the inputs were signed, the low two words of the 64 bit result is the signed result.

LongDivide

Function number = \$0D

Input	LongWord	Space for Remainder
Input	LongWord	Space for Quotient
Input	LongWord	Numerator
Input	LongWord	Denominator
Output	LongWord	Remainder
Output	LongWord	Quotient

Takes the two 32 bit unsigned inputs and divides them producing two 32 bit unsigned results.

FracDiv

Function number = \$12

Input	LongWord	Space for Result
Input	LongWord	Quotient
Input	LongWord	Divisor
Output	LongWord	Result

Divides two Frac inputs and returns a rounded Frac result (no remainder). Overflows return the most positive or negative value depending on the exclusive or of the inputs signs.

FixRound

Function number = \$13

Input	Word	Space for Result
Input	LongWord	Original Fixed value
Output	word	integer Result

Takes a Fixed input and returns a rounded integer result.

FracSqrt

Function number = \$14

Input	LongWord	Space for Result
Input	LongWord	Original Frac value
Output	LongWord	Result

Takes a Frac input and returns a rounded Frac square root. (Note: the input is taken as unsigned with the leading bit significant, i.e., the input range is from 0 to almost 4).

FracCos

Function number = \$15

Input	LongWord	Space for Result
Input	LongWord	Angle (fixed)
Output	LongWord	Result (fixed)

Takes a Fixed input (radians) and returns its cosine.

FracSin

Function number = \$16

Input	LongWord	Space for Result
Input	LongWord	Angle (fixed)
Output	LongWord	Result (fixed)

Takes a Fixed input (radians) and returns its sine.

FixATan2

Function number = \$17

Input	LongWord	Space for Result
Input	LongWord	Input 1
Input	LongWord	Input 2
Output	LongWord	Result (fixed)

Takes two inputs and returns a fixed point arc tangent (radians) of their coordinate. The inputs can be long integer, fixed or Frac (but must be of the same type).

HiWord

Function number = \$18

Input	Word	Space for Result
Input	LongWord	Input
Output	Word	Result

Returns high word of input.

LoWord

Function number = \$19

Input	Word	Space for Result
Input	LongWord	Input
Output	Word	Result

Returns low word of input.

Long2Fix

Function number = \$1A

Input	LongWord	Space for Result
Input	LongWord	Input
Output	LongWord	Result (fixed)

Converts long integer to fixed. Overflows return the most positive or negative value depending on the input sign.

Fix2Long

Function number = \$1B

Input	LongWord	Space for Result
Input	LongWord	Input
Output	LongWord	Result (LongInt)

Converts fixed to long integer. Conversions are rounded.

Fix2Frac

Function number = \$1C

Input	LongWord	Space for Result
Input	LongWord	Input
Output	LongWord	Result (Frac)

Converts fixed to Frac. Overflows return the most positive or negative value depending on the input sign.

Frac2Fix

Function number = \$1D

Input	LongWord	Space for Result
Input	LongWord	Input
Output	LongWord	Result (Fixed)

Converts Frac to Fixed. Conversions are rounded.

Fix2X

Function number = \$1E

Input	LongWord	Fixed value
Input	LongWord	Pointer to Extended

Converts Fixed to extended.

Frac2X

Function number = \$1F

Input	LongWord	Frac value
Input	LongWord	Pointer to Extended

Converts Frac to extended.

X2Fix

Function number = \$20

Input	LongWord	space for fixed result
Input	LongWord	Pointer to Extended
Output	LongWord	fixed result

Converts extended to Fixed. Conversions are rounded. Overflows, NaNs, and Infinities return the most positive or negative value depending on sign of the input.

X2Frac

Function number = \$21

Input	LongWord	space for frac result
Input	LongWord	Pointer to Extended
Output	LongWord	frac result

Converts extended to Frac. Conversions are rounded. Overflows, NaNs, and Infinities return the most positive or negative value depending on sign of the input.

CONVERSION ROUTINES

These routines convert between a binary value and an ASCII character string representing that value. The binary value can be either a 2-byte integer or a 4-byte integer. The character string can be in either hexadecimal or decimal format.

Int2Hex	Function number = \$22	
Input	Word	2-byte unsigned integer
Input	LongWord	Pointer to output string
Input	Word	Length of output string

Takes a 2-byte unsigned integer and produces an ASCII string representing the value in hexadecimal format. The string is right-justified and padded at the left with zeros. If the string is too short to represent the value, an error is returned. The ASCII characters in the output string have the high bit clear.

Long2Hex	Function number = \$23	
Input	LongWord	4-byte unsigned integer
Input	LongWord	Pointer to output string
Input	Word	Length of output string

Takes a 4-byte unsigned integer and produces an ASCII string representing the value in hexadecimal format. The string is right-justified and padded at the left with zeros. If the string is too short to represent the value, an error is returned. The ASCII characters in the output string have the high bit clear.

Hex2Int	Function number = \$24	
Input	Word	Space for result
Input	LongWord	Pointer to input string
Input	Word	Length of input string
Output	Word	2-byte unsigned integer

Takes an ASCII string representing a hexadecimal value and produces a 2-byte unsigned integer. The string should be right-justified and may be padded at the left with blanks or zeros. The ASCII characters in the string

may have the high bit either set or clear. Illegal characters in the string will cause an error to be returned. If the hexadecimal value is greater than \$FFFF, an overflow error will be returned.

Hex2Long

Function number = \$25

Input	LongWord	Space for Result
Input	LongWord	Pointer to input string
Input	Word	Length of input string
Output	LongWord	4-byte unsigned integer

Takes an ASCII string representing a hexadecimal value and produces a 4-byte unsigned integer. The string should be right-justified and may be padded at the left with blanks or zeros. The ASCII characters in the string may have the high bit either set or clear. Illegal characters in the string will cause an error to be returned. If the hexadecimal value is greater than \$FFFFFFFF, an overflow error will be returned.

Int2Dec

Function number = \$26

Input	Word	2-byte integer
Input	LongWord	Pointer to output string
Input	Word	Length of output string
Input	Word	Signed flag

Takes a 2-byte integer and produces an ASCII string representing the value in decimal format. The string is right-justified and padded at the left with blanks. The ASCII characters in the string have the high bit clear. If the Signed flag = 0, the integer will be considered to be unsigned. If the Signed flag <> 0, the integer will be considered to be signed. If a signed integer is negative, the string will contain an ASCII minus sign to the left of the most-significant digit. If the string is too short to represent the value, an error is returned.

Long2Dec

Function number = \$27

Input	LongWord	4-byte integer
Input	LongWord	Pointer to output string
Input	Word	Length of output string
Input	Word	Signed flag

Takes a 4-byte integer and produces an ASCII string representing the value in decimal format. The string is right-justified and padded at the left

with blanks. The ASCII characters in the string have the high bit clear. If the Signed flag = 0, the integer will be considered to be unsigned. If the Signed flag \neq 0, the integer will be considered to be signed. If a signed integer is negative, the string will contain an ASCII minus sign to the left of the most-significant digit. If the string is too short to represent the value, an error is returned.

Dec2Int

Function number = \$28

Input	Word	Space for result
Input	LongWord	Pointer to input string
Input	Word	Length of input string
Input	Word	Signed flag
Output	Word	2-byte integer

Takes an ASCII string representing a decimal value and produces a 2-byte integer. The string should be right-justified and may be padded at the left with blanks or zeros. The ASCII characters in the string may have the high bit either set or clear. If the Signed flag = 0, the value will be considered to be unsigned. If the Signed flag \neq 0, the value will be considered to be signed. If the value is signed, the string may contain an ASCII plus or minus sign directly in front of the most-significant digit. Illegal characters in the string will cause an error to be returned. If a signed value is greater than 32,767 or less than -32,768 an overflow error will be returned. If an unsigned value is greater than 65,535 an overflow error will be returned.

Dec2Long

Function number = \$29

Input	LongWord	Space for Result
Input	LongWord	Pointer to input string
Input	Word	Length of input string
Input	Word	Signed flag
Output	LongWord	4-byte integer

Takes an ASCII string representing a decimal value and produces a 4-byte integer. The string should be right-justified and may be padded at the left with blanks or zeros. The ASCII characters in the string may have the high bit either set or clear. If the Signed flag = 0, the value will be considered to be unsigned. If the Signed flag \neq 0, the value will be considered to be signed. If the value is signed, the string may contain an ASCII plus or minus sign directly in front of the most-significant digit. Illegal characters in the string will cause an error to be returned. If a signed value is greater than 2,147,483,647 or less than -2,147,483,648

an overflow error will be returned. If an unsigned value is greater than 4,294,967,295 an overflow error will be returned.

Hexlt

Function number = \$2A

Input	LongWord	Space for result
Input	Word	2-byte unsigned integer
Output	LongWord	4-byte hexadecimal string

Takes a 2-byte unsigned integer and returns a 4-byte ASCII string representing the value in hexadecimal format.

ERROR CODES

\$0B01	Bad input parameter
\$0B02	Illegal character in string
\$0B03	Integer or Long Integer overflow
\$0B04	String overflow