THE VALUE CONVERTER

INTRODUCTION

This chapter explains Resorcerer's Value Converter tool, which lets you view and convert among any of a dozen standard 32-bit Mac programming types. It behaves much like a Desk Accessory, only it is private to Resorcerer.

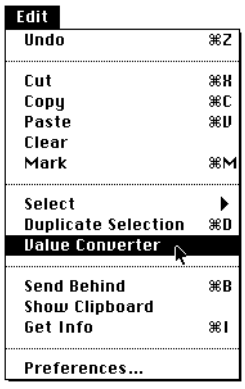
The Value Converter does not require any file or resource to be open in order to use it. It tends to come in the handiest, however, as an accessory to the Hex and Data Editors.

TOPICS COVERED

- Opening the Value Converter
- Window layout
- Operating the Value Converter
- Hot-linking to a Hex Selection
- Converter types

OPENING THE VALUE CONVERTER

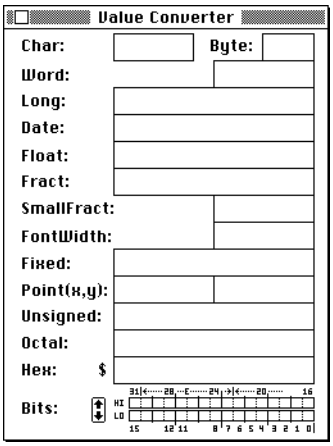
To open the Value Converter accessory, choose **Value Converter** from the **Edit** menu. If the Converter is already open, Resorcerer will bring it to the front of any other windows.



To close the Value Converter when you no longer need it, click in its Go Away box.

Sorcery: ⌘W will close the Value Converter window also.

WINDOW LAYOUT

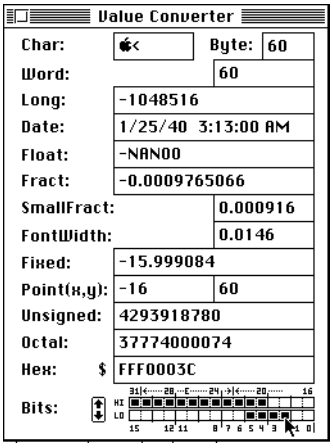


The Value Converter is a mini-editor for a single 32-bit long word. The 32 bits are displayed in two rows at the bottom of the Converter window. The top row, labeled **HI** on the left, shows the high-order 16 bits of the long word; the bottom row, labeled **LO**, shows the low-order 16 bits.

Each bit is depicted by a box that is either empty or filled with a black square. An empty box means the bit is clear; a black box means the bit is set. Individual bit numbers are also displayed above and below the bits, along with the range of bits that are used to encode the exponent in a single-precision (32-bit) floating point quantity.

Above the 32 bits are a series of editing boxes that hold various conversions of the 32 bits into different standard Macintosh data types. Each box is labeled on the left with its type. All of these conversions are presented to you as editable text. These different boxes let you view your 32-bit quantity, or portions of it, in various different ways simultaneously.

OPERATING THE VALUE CONVERTER



To set one or more bits, click in any empty box and drag the mouse over those bits you want to set. To clear one or more bits, click on any filled black box and drag the mouse over those bits you want to clear. To set or clear individual bits, just click and let go of the mouse directly over that bit, without dragging.

When you let go of the mouse button after settting or clearing one or more bits, the Value Converter converts the new 32-bit value into all of the types represented by the editing boxes in the rest of the dialog, and fills the edit fields in.

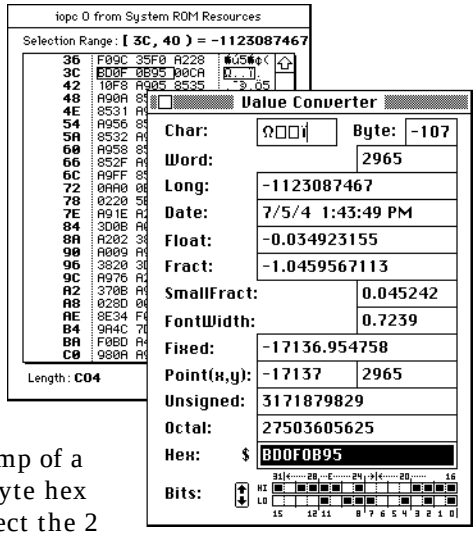
Next to the 32 bits is a bump-by-1 control. Click on the up or down arrow to increment or decrement the current 32-bit number. The speed at which the addition or subtraction occurs increases with the length of time you hold the mouse button.

Sorcery: To multiply or divide the 32-bit integer by 2 (that is, left- or right-shift it by 1), hold the Shift key down while clicking on an arrow.

When you change the text of any edit field, either by typing, cutting, pasting, or deleting characters, the 32-bit value for the changed value is recomputed and placed in the bit boxes below. The Value Converter then decodes the 32-bit quantity into all the other formats.

HOT-LINKING TO A HEX SELECTION

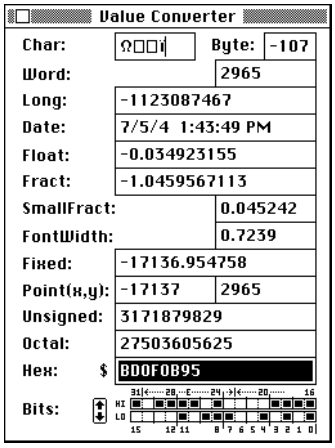
Whenever the Value Converter window becomes the frontmost active window, it looks at the window behind it to see what type of editing window it is. If that window is a Hex Editor window, or any other window that uses Resorcerer's hex editing package, such as the ones the Data Editor uses to edit hex fields, then you can use the Value Converter to help edit small portions of the hex data in any of the formats available.



For example, suppose that in a hex dump of a resource, you want to know what a 2-byte hex value looks like in binary or octal. Select the 2 bytes, and then bring the Value Converter to the front. It will install the current value of the 16 bits into the low-order half of the 32 bits it edits, and convert the 2 bytes into the various simultaneous formats. When you make any change to the value in any of the conversion fields, the change is immediately exported back to the hot-linked hex editing window, where the data changes automatically.

This lets you edit any 1-, 2-, or 4-byte selection in any hex editing window in any format the Value Converter supports.

CONVERTER TYPES



If you are running under System 7, you can turn on Balloon Help to show you important boundary conditions as you are editing.

CHARACTER

The 32 bits are divided into 4 consecutive 8-bit bytes, each of which is taken as the ASCII code for a character in the current system font. These four characters are then drawn into the **Char** field. The leftmost character is the high order byte; the rightmost character is the low order byte. However, leading null bytes (all 0 bits) are ignored. If there are more than four characters in the edit box, all but the final four are ignored.

To see the decimal equivalent of a single character, enter the single character into the **Char** field, and then look at the **Byte** field to see the value in decimal, the **Octal** field to see it in octal, the **Hex** field to see it in hex, and the **Bits** field to see it in binary.

BYTE

The **Byte** field displays the signed decimal value of the low order byte of your 32-bit quantity. If you attempt to enter a number outside the range of [-128,127], which implies that bits outside of the low order byte would have to be set, then the Converter will throw those bits away.

WORD

The **Word** field displays the signed decimal value of the low order word of your 32-bit quantity. If you attempt to enter a number outside the range of [-32768,32767], which implies that bits outside of the low order word have to be set, then the Converter will throw those bits away.

LONG

The **Long** field displays the signed decimal value of the entire 32-bit quantity.

DATE

The Date field takes the 32-bit number as the time elapsed in seconds since Jan. 1, 1904, and converts it to a standard date string.

FLOAT

The **Float** field displays the single-precision floating point value of your 32-bit quantity. The quantity is converted to a double prior to being converted to a string, which is why more digits than are actually significant are sometimes displayed.

Clicking on the least significant bit in the **Bits** field will give you a good feel for the numerical resolution of single-precision floats.

Note: Certain bit patterns are not legal floating point quantities. In these cases, the field may contain strange symbols such as "?", "N", or "NAN" (the last of these is an acronym for "Not A Number"). These symbols are created by SANE, Apple's numerics library that the Value Converter relies on to do its floating point conversions.

FRACT

The **Fract** field displays the signed `Fract` fixed point value of your 32-bit quantity. The high-order 2 bits represent the sign bit and the integer part of the number. The low-order 30 bits are the high-precision fractional part. Fract numbers can encode fixed point numbers between 2.0 and -2.0.

SMALLFRACT

The **SmallFract** field displays the `SmallFract` fixed point value of the low-order 16 bits of your 32-bit quantity. `SmallFract`s are unsigned numbers between 0.0 and 1.0 used to encode color components. A `SmallFract` is the fractional part of a `Fixed` number.

FONTWIDTH

The **FontWidth** field displays the low-order word of the 32-bits as 16-bit fixed point value where the upper 4 bits of the word are the sign bit and integer part of the number and the lower 12 bits of the word are the fractional part.

FIXED POINT

The **Fixed Point** field displays your 32-bit quantity in `Fixed` format. The high order 16-bit word represents the sign bit and the integer part of the number; the low order 16-bit word encodes the fractional part of the fixed point quantity (16:16 format)

POINT

The **Point** field displays both the X (horizontal) and Y (vertical) values of a Quickdraw `Point` record, where the high order word of the 32-bit quantity is the signed Y coordinate, and the low order word is the signed X coordinate.

UNSIGNED DECIMAL

The **Unsigned** field displays your 32-bit quantity as an unsigned decimal number, which will be the same value as the signed **Long** field except when the high order sign bit (bit 31) is set. If you enter a signed decimal number, it will be replaced by the unsigned version of the same 32 bits.

UNSIGNED OCTAL

The **Octal** field displays your 32-bit quantity as an unsigned octal number. The field is often useful for PostScript programming and debugging, since PostScript programs often use octal encodings for character sets. Leading 0's are suppressed.

UNSIGNED HEX

The **Hex** field displays your 32-bit quantity as an unsigned hexadecimal number. Each pair of hex digits represents one of the four bytes in the 32-bits. Leading 0's are suppressed.

BINARY

The **Bits** field displays each bit of your 32-bit quantity as black if set, white if clear. The individual bit numbers are labeled, as well as the exponent bits (bits 23 - 30) of a single precision 32-bit float.