**What is a Hex Editor?**

|  |
| --- |
| [Text versus Binary](http://www.sweetscape.com/graphics/ArtTextBinary.GIF) *Figure 1 - Shows a text file loaded in notepad on the left (human readable) versus a binary file on the right.* |

A hex editor is a computer program used to view and edit binary files. A binary file is a file that contains data in machine-readable form (as apposed to a text file which can be read by a human - see Figure 1). Hex editors allow editing the raw data contents of a file, instead of other programs which attempt to interpret the data for you. Since a hex editor is used to edit binary files, they are sometimes called abinary editor or a binary file editor. If you edit a file with a hex editor, you are said to hex edit the file, and the process of using a hex editor is called hex editing.

A typical hex editor has three areas: an address area on the left, a hexadecimal area in the center, and a character areaon the right (see Figure 2). In the most powerful hex editors, these areas can be configured to display other values. Every file on your machine is made up of a series of bytes. A byte is just a number that can store a value from 0 up to 255 inclusive. Every byte in a file is assigned a number, called its address, starting at 0 for the first byte of the file, 1 for the second byte, etc.

|  |  |  |
| --- | --- | --- |
| |  | | --- | | [Hex Editor](http://www.sweetscape.com/graphics/ArtHexEditor.GIF) *Figure 2 - Shows a hex editor with addresses on the left, a hexadecimal area in the middle, and a character area on the right.* | |  |

**Hexadecimal Area**

The middle hexadecimal area is the most commonly used area of a hex editor. It lists each byte of the file in a table, usually sixteen bytes per line. If Figure 2, the first 8 bytes of the file would be '4D 5A 90 00 03 00 00 00'. Each byte is listed in a special notation called hexadecimal notation, sometimes called hex code. Most numbers in our world are represented in decimal or base-10 notation, meaning we use 10 different digits (0 up to 9). Hexadecimal or base-16 notation uses 16 different digits: 0 up to 9 and then the letters A, B, C, D, E, F to represent 10, 11, 12, 13, 14, and 15. Numbers in hexadecimal notation commonly have an 'h' written after them, or an '0x' written before them to indicate that they are in base-16 notation (for example: 1Fh or 0xA7). To convert a 2-digit hexadecimal number to a regular number, multiply the first digit by 16 and add it to the second digit. For example, 3A = 3\*16 + 10 = 58.

**Character Area**

Although each byte in a file can store a value from 0 to 255, what matters is what this data means. One way bytes are used is to assign a different letter or symbol to each possible value. For example, the byte value 65 could represent the character 'A' and the value 33 could represent the symbol '!'. The byte values 0 to 127 are usually assigned letters and symbols according to a standard called ASCII (although other standards exist). The character area on the right of the hex editor displays the ASCII representation of each of the bytes in the hexadecimal area. For example in Figure 2, the second byte of the file (5A) is displayed as 'Z' in the character area. Some byte values represent special codes that cannot be displayed in the character area. If there is no character that can be displayed, usually a '.' is displayed in the hex editor.

**Address Area**

The address area on the left side of the hex editor displays the address of the first byte of each line. The addresses are usually displayed in hexadecimal format, but many hex editors can display addresses in decimal format as well. In Figure 2, the address of the first byte of the first line would be 0, and the address of the first byte of the second line would be 0010h (or 16 in decimal format). Remember that addresses start from 0 so if you have a file of 512 bytes, the byte addresses would range from 0 up to 511. Some hex editors also have a ruler along the top of the hex editor to help you know the addresses of other bytes in the file (the orange bar in Figure 2).

**Editing Data**

Data can be edited in a hex editor just like a normal text editor. A hex editor has a cursor that can be moved by clicking with the mouse or using the cursor keys. Position the cursor over the byte you want to edit, and type the value you want to change to using the keyboard. The cursor can be switched between the hexadecimal area and the character area by pressing the 'Tab' key. When the cursor is in the hexadecimal area, you have to enter byte values in hexadecimal notation, but when the cursor is in the character area, you can enter regular characters just like a text editor. Good hex editors always have an Overwrite mode and an Insert mode. In Overwrite mode, typing values on the keyboard just changes the existing byte values, but in Insert mode, typing on the keyboard inserts new bytes into the file. You can switch between Insert and Overwrite mode by pressing the 'Ins' key. Data can also be edited by selecting a set of bytes. Selections are made just like a text editor: click and drag the mouse or hold the 'Shift' key while moving the cursor. Once a selection has been made, the file can be edited using the standard Cut, Copy, and Paste commands on the Edit menu.

**Advanced Features**

The best hex editors in the world contain a whole host of other tools to help you work with binary files. For example, you could use a Find tool to locate certain bytes in a file, or a Binary Comparison tool to compare binary bytes between two files. Some hex editors can even edit the bytes of hard drives and processes just like you would a binary file. The most advanced feature of hex editors is now the ability to place a template over a file that allow you to understand what the bytes of a binary file actually mean.