# Investigating File Formats

### Aim:

 To use a Hex Editor to investigate the binary representations within different file types

### Objectives:

1. Learn to use a hex editor
2. Investigate text files
3. Investigate BMP image files
4. Learn how VB.net writes binary files.

### Getting Started

Download HexEditorAndDataRepFiles.zip from the “Resources and Portable Software” Folder on GodalmingOnline.

There are LOTS of youTube vids on how to use a Hex editor… Usually hacking games in some form or other… Hex editors change the binary values in a file, disk or process!!! You can totally destroy data VERY easily. Forums are full of people saying… *“ I tried X, now my hard dive isn’t recognised anymore… What Happened?”* **BE CAREFUL.. BE AFRAID!**

## Learning to use a Hex editor.

## http://imag.malavida.com/mvimgbig/download/hxd-8730-1.jpg

ASCII Chars

HEX Codes

Start Position

The First Hex Code in the above file is 56 to calculate the decimal value … 6x1 + 5x16 = 86. The Ascii character for the value 86 is V (capital V) … The ascii table at the end of this document have the Hex Codes so you don’t need to calculate the Decimal value… OBVIOUSLY the actual binary value in the file is: 0101 0110 !

## Investigate Text Files

* Open the file ***“Sample1.txt”*** using HxD. ALSO open it use NotePad (or wordpad or even word)
* Check that the hex codes make sense to you… What do the codes 0D 0A mean?
* *In HxD* :Change one code to change a character.. Save the file (ignore any warnings!)
Reopen the file in Notepad.. Hopefully it has changed!

## Investigating bitmaps

BitMaps have a file header which include File Size, width height, colour depth etc.. For the full Specification for Windows Bitmaps follow this link: <http://www.dragonwins.com/domains/getteched/bmp/bmpfileformat.htm>

* Load the various BMP examples from the zip folder into HxD (the filenames should give you a clue as to what they are!) Open them in msPAINT .. You can see that the file header has a few small changes.. see if you can spot what the values are! YOU DO NOT NEED THIS LEVEL OF DETAIL FOR THE AS EXAM. You do need to know there is a file header… and that pixels are represented by binary words (which may be more than one set of hex codes!)
* (NB.. I would have chosen an 8bit format but it appears it’s not supported!)
* Load 1pxBMPsample16bit Have a go at changing the pixel (the last hex code in the file).. Check in paint that the pixel has changed!
* Load 100pxBMPsample16bit into HxD Can you make the face happy rather than sad? DON’T use paint to edit the file.. use windows image viewer to see how you are doing. (F5 refreshes).

## Learn how VB.net writes Binary Files

The list of VB Data Types and the storage allocation (how many bytes that data type uses) is at the end of this document.

Create a new VB Project called BinaryWriteTest copy the following code into the Main Program and hit save all.

Dim testInt as integer = 9
Dim testChar as Char = “a”
Dim testStr as String =”bob”

FileOpen(1, “BinaryTest1.bin”, OpenMode.Binary, OpenAccess.Write)
fileput(1,testInt)
FileClose(1)

### Integers

* Run the program… Now open “BinaryTest1.bin” using HxD.
* How many bytes have been written to the file?
* Change the assignment to 10, Run the code, refresh Hxd. Do you understand what has happened?
* Change the assignment to 17, Run the code, refresh Hxd. Do you understand what has happened?
* Change the assignment to 255, then 256, then 4095 then 4096 Check the Hex each time.. are you 100% sure how the Hex digits are been placed!!! Try 2,147,483,647.. do you find anything surprising about the hex value.. Check the VB data table..
* Do the ASCII characters have any meaning at all to this experiment?

### Char

* Alter your code: Change the file to “TestBin2.bin” change the parameter in the FilePut call to testChar
* Run your code and Open TestBin2.bin with HxD . Do you understand what the codes mean?
* Change the assignment to “b”, Run the code, refresh Hxd. Have you noticed that the Text section helps you now?

### Structures

The nice thing about binary writing is that you can write a whole structure in one go…

Structure MyData

Dim testInt as integer
Dim testChar as Char
Dim testStr as String

End structure

Dim TestStruc as MyData

TestStruc.testInt = 9
TestStruc.testChar = “a”
TestStruc.testStr =”bob”

FileOpen(1, “BinaryTest5.bin”, OpenMode.Binary, OpenAccess.Write)
fileput(1,TestStruc)
FileClose(1)

When strings as being combined with other data we need to know how long they are… change the values in the code and work out how they are being written in HxD. Try to change the values in HxD.

Alter your code to read the fine and output the values.

How doe VB record how long a string is in a structure?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dec** | **Hex** | **Char** | **Name** |  | **Dec** | **Hex** | **Char Zoom...** | **Name** |
| 0 | 00 | NUL | Null |  | 128 | 80 | **€** | Euro Sign |
| 1 | 01 | STX | Start of Header |  | 129 | 81 |  | Undefined |
| 2 | 02 | SOT | Start of Text |  | 130 | 82 | **‚** | Single Low-9 Quotation Mark |
| 3 | 03 | ETX | End of Text |  | 131 | 83 | **ƒ** | Latin Small Letter F With Hook |
| 4 | 04 | EOT | End of Transmission |  | 132 | 84 | **„** | Double Low-9 Quotation Mark |
| 5 | 05 | ENQ | Enquiry |  | 133 | 85 | **…** | Horizontal Ellipsis |
| 6 | 06 | ACK | Acknowledge |  | 134 | 86 | **†** | Dagger |
| 7 | 07 | BEL | Bell |  | 135 | 87 | **‡** | Double Dagger |
| 8 | 08 | BS | BackSpace |  | 136 | 88 | **ˆ** | Modifier Letter Circumflex Accent |
| 9 | 09 | HT | Horizontal Tabulation |  | 137 | 89 | **‰** | Per Mille Sign |
| 10 | 0A | LF | Line Feed |  | 138 | 8A | **Š** | Latin Capital Letter S With Caron |
| 11 | 0B | VT | Vertical Tabulation |  | 139 | 8B | **‹** | Single Left-pointing Angle Quotation Mark |
| 12 | 0C | FF | Form Feed |  | 140 | 8C | **Œ** | Latin Capital Ligature Oe |
| 13 | 0D | CR | Carriage Return |  | 141 | 8D |  | Undefined |
| 14 | 0E | SO | Shift Out |  | 142 | 8E | **Ž** | Latin Capital Letter Z With Caron |
| 15 | 0F | SI | Shift In |  | 143 | 8F |  | Undefined |
| 16 | 10 | DLE | Data Link Escape |  | 144 | 90 |  | Undefined |
| 17 | 11 | DC1 | Device Control 1 (XON) |  | 145 | 91 | **‘** | Left Single Quotation Mark |
| 18 | 12 | DC2 | Device Control 2 |  | 146 | 92 | **’** | Right Single Quotation Mark |
| 19 | 13 | DC3 | Device Control 3 (XOFF) |  | 147 | 93 | **“** | Left Double Quotation Mark |
| 20 | 14 | DC4 | Device Control 4 |  | 148 | 94 | **”** | Right Double Quotation Mark |
| 21 | 15 | NAK | Negative acknowledge |  | 149 | 95 | **•** | Bullet |
| 22 | 16 | SYN | Synchronous Idle |  | 150 | 96 | **–** | En Dash |
| 23 | 17 | ETB | End of Transmission Block |  | 151 | 97 | **—** | Em Dash |
| 24 | 18 | CAN | Cancel |  | 152 | 98 | **˜** | Small Tilde |
| 25 | 19 | EM | End of Medium |  | 153 | 99 | **™** | Trade Mark Sign |
| 26 | 1A | SUB | Substitute |  | 154 | 9A | **š** | Latin Small Letter S With Caron |
| 27 | 1B | ESC | Escape |  | 155 | 9B | **›** | Single Right-pointing Angle Quotation Mark |
| 28 | 1C | FS | File Separator |  | 156 | 9C | **œ** | Latin Small Ligature Oe |
| 29 | 1D | GS | Group Separator |  | 157 | 9D |  | Undefined |
| 30 | 1E | RS | Record Separator |  | 158 | 9E | **ž** | Latin Small Letter Z With Caron |
| 31 | 1F | US | Unit Separator |  | 159 | 9F | **Ÿ** | Latin Capital Letter Y With Diaeresis |
| 32 | 20 | [Space] | Space |  | 160 | A0 |  | No-break Space |
| 33 | 21 | ! | Exclamation mark |  | 161 | A1 | **¡** | Inverted Exclamation Mark |
| 34 | 22 | " | Quotes |  | 162 | A2 | **¢** | Cent Sign |
| 35 | 23 | # | Hash |  | 163 | A3 | **£** | Pound Sign |
| 36 | 24 | $ | Dollar |  | 164 | A4 | **¤** | Currency Sign |
| 37 | 25 | % | Percent |  | 165 | A5 | **¥** | Yen Sign |
| 38 | 26 | & | Ampersand |  | 166 | A6 | **¦** | Broken Bar |
| 39 | 27 | ' | Apostrophe |  | 167 | A7 | **§** | Section Sign |
| 40 | 28 | ( | Open bracket |  | 168 | A8 | **¨** | Diaeresis |
| 41 | 29 | ) | Close bracket |  | 169 | A9 | **©** | Copyright Sign |
| 42 | 2A | \* | Asterisk |  | 170 | AA | **ª** | Feminine Ordinal Indicator |
| 43 | 2B | + | Plus |  | 171 | AB | **«** | Left-pointing Double Angle Quotation Mark |
| 44 | 2C | , | Comma |  | 172 | AC | **¬** | Not Sign |
| 45 | 2D | - | Dash |  | 173 | AD | **­** | Soft Hyphen |
| 46 | 2E | . | Full stop |  | 174 | AE | **®** | Registered Sign |
| 47 | 2F | / | Slash |  | 175 | AF | **¯** | Macron |
| 48 | 30 | 0 | Zero |  | 176 | B0 | **°** | Degree Sign |
| 49 | 31 | 1 | One |  | 177 | B1 | **±** | Plus-minus Sign |
| 50 | 32 | 2 | Two |  | 178 | B2 | **²** | Superscript Two |
| 51 | 33 | 3 | Three |  | 179 | B3 | **³** | Superscript Three |
| 52 | 34 | 4 | Four |  | 180 | B4 | **´** | Acute Accent |
| 53 | 35 | 5 | Five |  | 181 | B5 | **µ** | Micro Sign |
| 54 | 36 | 6 | Six |  | 182 | B6 | **¶** | Pilcrow Sign |
| 55 | 37 | 7 | Seven |  | 183 | B7 | **·** | Middle Dot |
| 56 | 38 | 8 | Eight |  | 184 | B8 | **¸** | Cedilla |
| 57 | 39 | 9 | Nine |  | 185 | B9 | **¹** | Superscript One |
| 58 | 3A | : | Colon |  | 186 | BA | **º** | Masculine Ordinal Indicator |
| 59 | 3B | ; | Semi-colon |  | 187 | BB | **»** | Right-pointing Double Angle Quotation Mark |
| 60 | 3C | < | Less than |  | 188 | BC | **¼** | Vulgar Fraction One Quarter |
| 61 | 3D | = | Equals |  | 189 | BD | **½** | Vulgar Fraction One Half |
| 62 | 3E | > | Greater than |  | 190 | BE | **¾** | Vulgar Fraction Three Quarters |
| 63 | 3F | ? | Question mark |  | 191 | BF | **¿** | Inverted Question Mark |
| 64 | 40 | @ | At |  | 192 | C0 | **À** | Latin Capital Letter A With Grave |
| 65 | 41 | A | Uppercase A |  | 193 | C1 | **Á** | Latin Capital Letter A With Acute |
| 66 | 42 | B | Uppercase B |  | 194 | C2 | **Â** | Latin Capital Letter A With Circumflex |
| 67 | 43 | C | Uppercase C |  | 195 | C3 | **Ã** | Latin Capital Letter A With Tilde |
| 68 | 44 | D | Uppercase D |  | 196 | C4 | **Ä** | Latin Capital Letter A With Diaeresis |
| 69 | 45 | E | Uppercase E |  | 197 | C5 | **Å** | Latin Capital Letter A With Ring Above |
| 70 | 46 | F | Uppercase F |  | 198 | C6 | **Æ** | Latin Capital Ligature Ae |
| 71 | 47 | G | Uppercase G |  | 199 | C7 | **Ç** | Latin Capital Letter C With Cedilla |
| 72 | 48 | H | Uppercase H |  | 200 | C8 | **È** | Latin Capital Letter E With Grave |
| 73 | 49 | I | Uppercase I |  | 201 | C9 | **É** | Latin Capital Letter E With Acute |
| 74 | 4A | J | Uppercase J |  | 202 | CA | **Ê** | Latin Capital Letter E With Circumflex |
| 75 | 4B | K | Uppercase K |  | 203 | CB | **Ë** | Latin Capital Letter E With Diaeresis |
| 76 | 4C | L | Uppercase L |  | 204 | CC | **Ì** | Latin Capital Letter I With Grave |
| 77 | 4D | M | Uppercase M |  | 205 | CD | **Í** | Latin Capital Letter I With Acute |
| 78 | 4E | N | Uppercase N |  | 206 | CE | **Î** | Latin Capital Letter I With Circumflex |
| 79 | 4F | O | Uppercase O |  | 207 | CF | **Ï** | Latin Capital Letter I With Diaeresis |
| 80 | 50 | P | Uppercase P |  | 208 | D0 | **Ð** | Latin Capital Letter Eth |
| 81 | 51 | Q | Uppercase Q |  | 209 | D1 | **Ñ** | Latin Capital Letter N With Tilde |
| 82 | 52 | R | Uppercase R |  | 210 | D2 | **Ò** | Latin Capital Letter O With Grave |
| 83 | 53 | S | Uppercase S |  | 211 | D3 | **Ó** | Latin Capital Letter O With Acute |
| 84 | 54 | T | Uppercase T |  | 212 | D4 | **Ô** | Latin Capital Letter O With Circumflex |
| 85 | 55 | U | Uppercase U |  | 213 | D5 | **Õ** | Latin Capital Letter O With Tilde |
| 86 | 56 | V | Uppercase V |  | 214 | D6 | **Ö** | Latin Capital Letter O With Diaeresis |
| 87 | 57 | W | Uppercase W |  | 215 | D7 | **×** | Multiplication Sign |
| 88 | 58 | X | Uppercase X |  | 216 | D8 | **Ø** | Latin Capital Letter O With Stroke |
| 89 | 59 | Y | Uppercase Y |  | 217 | D9 | **Ù** | Latin Capital Letter U With Grave |
| 90 | 5A | Z | Uppercase Z |  | 218 | DA | **Ú** | Latin Capital Letter U With Acute |
| 91 | 5B | [ | Open square bracket |  | 219 | DB | **Û** | Latin Capital Letter U With Circumflex |
| 92 | 5C | \ | Backslash |  | 220 | DC | **Ü** | Latin Capital Letter U With Diaeresis |
| 93 | 5D | ] | Close square bracket |  | 221 | DD | **Ý** | Latin Capital Letter Y With Acute |
| 94 | 5E | ^ | Caret / hat |  | 222 | DE | **Þ** | Latin Capital Letter Thorn |
| 95 | 5F | \_ | Underscore |  | 223 | DF | **ß** | Latin Small Letter Sharp S |
| 96 | 60 | ` | Grave accent |  | 224 | E0 | **à** | Latin Small Letter A With Grave |
| 97 | 61 | a | Lowercase a |  | 225 | E1 | **á** | Latin Small Letter A With Acute |
| 98 | 62 | b | Lowercase b |  | 226 | E2 | **â** | Latin Small Letter A With Circumflex |
| 99 | 63 | c | Lowercase c |  | 227 | E3 | **ã** | Latin Small Letter A With Tilde |
| 100 | 64 | d | Lowercase d |  | 228 | E4 | **ä** | Latin Small Letter A With Diaeresis |
| 101 | 65 | e | Lowercase e |  | 229 | E5 | **å** | Latin Small Letter A With Ring Above |
| 102 | 66 | f | Lowercase f |  | 230 | E6 | **æ** | Latin Small Ligature Ae |
| 103 | 67 | g | Lowercase g |  | 231 | E7 | **ç** | Latin Small Letter C With Cedilla |
| 104 | 68 | h | Lowercase h |  | 232 | E8 | **è** | Latin Small Letter E With Grave |
| 105 | 69 | i | Lowercase i |  | 233 | E9 | **é** | Latin Small Letter E With Acute |
| 106 | 6A | j | Lowercase j |  | 234 | EA | **ê** | Latin Small Letter E With Circumflex |
| 107 | 6B | k | Lowercase k |  | 235 | EB | **ë** | Latin Small Letter E With Diaeresis |
| 108 | 6C | l | Lowercase l |  | 236 | EC | **ì** | Latin Small Letter I With Grave |
| 109 | 6D | m | Lowercase m |  | 237 | ED | **í** | Latin Small Letter I With Acute |
| 110 | 6E | n | Lowercase n |  | 238 | EE | **î** | Latin Small Letter I With Circumflex |
| 111 | 6F | o | Lowercase o |  | 239 | EF | **ï** | Latin Small Letter I With Diaeresis |
| 112 | 70 | p | Lowercase p |  | 240 | F0 | **ð** | Latin Small Letter Eth |
| 113 | 71 | q | Lowercase q |  | 241 | F1 | **ñ** | Latin Small Letter N With Tilde |
| 114 | 72 | r | Lowercase r |  | 242 | F2 | **ò** | Latin Small Letter O With Grave |
| 115 | 73 | s | Lowercase s |  | 243 | F3 | **ó** | Latin Small Letter O With Acute |
| 116 | 74 | t | Lowercase t |  | 244 | F4 | **ô** | Latin Small Letter O With Circumflex |
| 117 | 75 | u | Lowercase u |  | 245 | F5 | **õ** | Latin Small Letter O With Tilde |
| 118 | 76 | v | Lowercase v |  | 246 | F6 | **ö** | Latin Small Letter O With Diaeresis |
| 119 | 77 | w | Lowercase w |  | 247 | F7 | **÷** | Division Sign |
| 120 | 78 | x | Lowercase x |  | 248 | F8 | **ø** | Latin Small Letter O With Stroke |
| 121 | 79 | y | Lowercase y |  | 249 | F9 | **ù** | Latin Small Letter U With Grave |
| 122 | 7A | z | Lowercase z |  | 250 | FA | **ú** | Latin Small Letter U With Acute |
| 123 | 7B | { | Open brace |  | 251 | FB | **û** | Latin Small Letter U With Circumflex |
| 124 | 7C | | | Pipe |  | 252 | FC | **ü** | Latin Small Letter U With Diaeresis |
| 125 | 7D | } | Close brace |  | 253 | FD | **ý** | Latin Small Letter Y With Acute |
| 126 | 7E | ~ | Tilde |  | 254 | FE | **þ** | Latin Small Letter Thorn |
| 127 | 7F | DEL | Delete |  | 255 | FF | **ÿ** | Latin Small Letter Y With Diaeresis |

|  |  |  |
| --- | --- | --- |
| Visual Basic type  | storage allocation  | Value range  |
| **Boolean**  | 2 bytes  | **True** or **False.**  |
| **Byte**  | 1 byte  | 0 through 255 (unsigned).  |
| **Char**  | 2 bytes  | 0 through 65535 (unsigned).  |
| **Date**  | 8 bytes  | 0:00:00 on January 1, 0001 through 11:59:59 PM on December 31, 9999.  |
| **Decimal**  | 16 bytes  | 0 through +/-79,228,162,514,264,337,593,543,950,335 with no decimal point; 0 through +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/-0.0000000000000000000000000001 (+/-1E-28).  |
| **Double** (double-precision) | 8 bytes  | -1.79769313486231570E+308 through -4.94065645841246544E-324 for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 for positive values.  |
| **Integer**  | 4 bytes  | -2,147,483,648 through 2,147,483,647.  |
| **Long** (long integer)  | 8 bytes  | -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807.  |
| **Short**  | 2 bytes  | -32,768 through 32,767.  |
| **Single**  | 4 bytes  | -3.4028235E+38 through -1.401298E-45 for negative values; 1.401298E-45 through 3.4028235E+38 for positive values.  |
| **String** (variable-length)  | Depends | 0 to approximately 2 billion Unicode characters.  |
| **User-Defined Type** (structure)  | Depends | Each member of the structure has a range determined by its data type and independent of the ranges of the other members.  |