

# ITFreeTraining



## Converting Decimals

For the free video please see  
<http://itfreetraining.com/ipv4/converting-decimals>

In this video from ITFreeTraining, I will look at how to convert between decimals, binary and hexadecimal. This is a skill that is required for working with IPv4 and IPv6.

Access the rest of the course at <http://ITFreeTraining.com/ipv4>

Download the PDF handout at <http://ITFreeTraining.com/handouts/ipv4/converting-decimals.pdf>

# In This Video

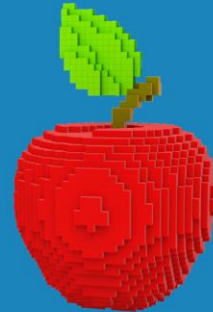
Binary



Hexadecimal

7B	B2	3A	2F	5E	02	1F	3D	BB	2E	62	A4	7A
5E	C9	10	96	37	34	22	92	83	E3	2D	3D	60
B5	47	75	52	FB	7A	26	E9	46	8C	77	4D	12
9D	46	C8	06	DD	05	4A	55	53	7E	AC	2C	DF
24	06	78	2F	D6	9D	19	E0	3C	BA	34	B3	74
E1	F6	82	7D	28	B5	6D	1E	D6	F6	F6	C9	53
7E	D8	24	38	B4	02	24	F6	35	04	5B	49	24
E9	B4	DF	B4	9B	4A	40	FE	6C	E0	6D	04	AD
54	2C	5F	ED	CE	B6	D8	78	D6	36	CD	85	B0
DB	E6	BE	87	5E	62	76	59	B5	A5	59	62	2C
CC	95	67	9D	1E	B9	B1	C2	88	F8	82	B7	8D

Bytes & Others



0:09 – This video looks at binary, hexadecimal, and how they relate to bytes and other units. I will look at how you would convert between these. This skill becomes particularly important when designing networks and is a must have for the administrator.

# What is binary?

- A value/switch that is either on or off
- Example binary string 01001110



**Value is 1**



**Value is 0**

0:27 – Binary is a system that uses values or switches that are either on or off. If the switch is on, the value is one. If the switch is off, the value is 0. This gives you two possible values. However, if you group multiple binary values together it is possible to represent larger numbers. A common way of grouping multiple binary digits together is by using a string. In this example, 8 binary values have been combined together in a string. An individual zero or one is known as a bit. In this example, there are 8 bits in the binary string. The next step to understanding binary is how would a binary string get converted to decimal?

# Power of 2

Power	Expression	Value
0	$2^0$	1
1	$2^1$	2
2	$2^2$	4
3	$2^3$	8
4	$2^4$	16
5	$2^5$	32
6	$2^6$	64
7	$2^7$	128

1:13 – When converting between different types of units, the power of 2 table is used. The first row is two to the power of zero. This gives a result of one. In computing, counting will often start with the value of zero. This means, two to the power of zero can give two values. These are zero and one.

If you consider the next row, two to the power of one gives a value of two. So the value has doubled. The next row of two to the power of two gives a value of four so it has doubled again. The power of two table becomes important when converting from binary to decimal or decimal to binary. For this reason, it is a good idea to memorize this table.

# Binary to Decimal

- Example string 10011010

	Column 8	Column 7	Column 6	Column 5	Column 4	Column 3	Column 2	Column 1
Powers of 2	128 (2 <sup>7</sup> )	64 (2 <sup>6</sup> )	32 (2 <sup>5</sup> )	16 (2 <sup>4</sup> )	8 (2 <sup>3</sup> )	4 (2 <sup>2</sup> )	2 (2 <sup>1</sup> )	1 (2 <sup>0</sup> )
Example	1	0	0	1	1	0	1	0
Value	128	0	0	16	8	0	2	0

$$\begin{aligned} \text{Result} &= 128 + 0 + 0 + 16 + 8 + 0 + 2 + 0 \\ &= 154 \end{aligned}$$

1:58 – To convert a binary string to decimal a conversion table is used. In this example, the binary string 10011010 will be converted to decimal. To do this, the binary string is added to the table going from right to left. The value at the bottom of the table is calculated by multiplying the row “Powers of 2” by the row “Example”. In the case of column 8, this would be 128\*1 which would give 128. In the case of column 7 64\*0 would give a result of 0. Columns 6, 3 and 1 are all 0 so would give a result of 0. Columns 5, 4 and 2 have 1 values so the results would be column 5 1\*16=16, column 4 1\*8=8 and column 2 1\*2=2.

To calculate the end result, it is just a matter of adding up all the results from the above. This would give 128+0+0+16+8+0+2+0 = 154. Thus the final result is 154.

# Decimal to Binary

- Example 154

	Column 8	Column 7	Column 6	Column 5	Column 4	Column 3	Column 2	Column 1
Powers of 2	128 (2 <sup>7</sup> )	64 (2 <sup>6</sup> )	32 (2 <sup>5</sup> )	16 (2 <sup>4</sup> )	8 (2 <sup>3</sup> )	4 (2 <sup>2</sup> )	2 (2 <sup>1</sup> )	1 (2 <sup>0</sup> )
Test	154 >= 128	Subtract 128 26 >= 64	26 > = 32	26 >= 16	Subtract 16 10 >= 8	Subtract 8 2 >= 4	2 >= 2	Subtract 2 0 >= 1
Result	1	0	0	1	1	0	1	0

Result = 1      0      0      1      1      0      1      0  
 = 10011010

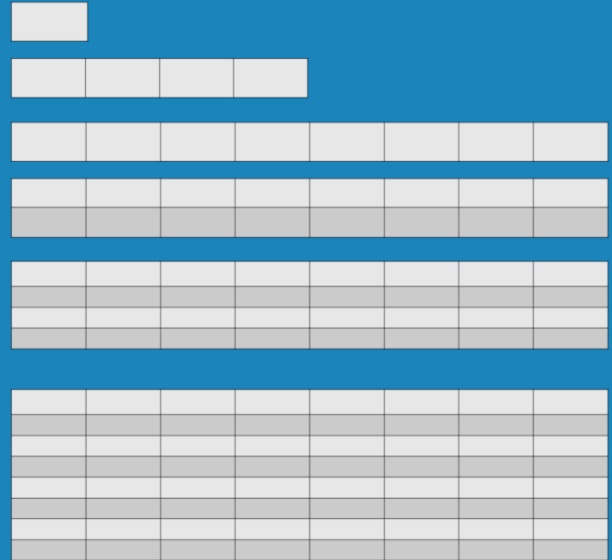
4:46 – In this example, the previous example of 154 will be converted to binary. This is done by using the powers of 2 table. In order to do this, the value is compared with the power of 2 table starting from the left moving right. Starting with column 8, 154 is compared with 128. 154 is >= 128 so a 1 is recorded. Since a 1 is recorded, 128 needs to be subtracted from 154 giving 26 which is placed in column 7. Then 26 is compared with 64, this is less than 64 so a 0 is recorded. Since a 0 was recorded the value is copied to the next column. That is, 26 is copied to column 6.

The process repeats for each column. Column 6, 26 is not >= 32 so 0 is recorded. Column 5, 26 is >= 16 so 1 is recorded. Since a 1 is recorded, 16 is subtracted from 26 giving 10. The result, 10 is recorded in column 4. Column 4, 10 is >= 8 so a 1 is recorded. Since a 1 was recorded, 8 is subtracted from 10 and the result of 2 is recorded in column 2. Column 2, 2 is >= 2 so a 1 is recorded. Since a 1 was recorded 2 is subtracted from 2 which gives a result of 0 which is recorded in column 1. Column 1, 0 is not >= 1 so a 0 is recorded.

Looking at the result row, this gives the binary value of 10011010.

# Bytes and Other Units

- 1 bit = Bit
- 4 bits = Nibble or Nybble
- 8 bits = Byte
- 16 bits = Word
- 32 bits = Long Word
- 64 bits = Very Long Word



7:25 – The most common data type you will most likely come across is a byte, but it does help to know the other ones.

Bit - This is the smallest data unit. Bits are grouped together to form other types.

Nibble or Nybble – This is 4 bits grouped together. A Nibble is useful when working with hexadecimal.

Byte – 8 bits grouped together is called a byte. Even though there are smaller types like nibbles and bits, a byte is often the smallest type that is used. For example, when writing to a file on a hard disk you will only be able to write data as multiples of bytes, you will not be able to write a single bit.

Word - This is 16 bits or two bytes. Sometimes you will use this in networking.

Long word - This is 4 bytes or 2 words. Since it is 2 words, it is sometimes referred to as a double word. With IPv4, you come across this data type since IPv4 addresses are 32 bits long.

Very long word – This is 64 bits. This is also referred to as a Quad Word. You will come across a data type of this size if you use IPv6, as a 128 bit IPv6 address is often broken into two 64bit parts.

# Hexadecimal

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Example value 154

1 0 0 1 1 0 1 0

Bits	1	0	0	1
Powers of 2	8	4	2	1

Result = 8 + 0 + 0 + 1  
 Decimal = 9  
 Hexadecimal = 9

Bits	1	0	1	0
Powers of 2	8	4	2	1

Result = 8 + 0 + 2 + 0  
 Decimal = 10  
 Hexadecimal = A

154 is 9A in Hexadecimal

9:00 – Hexadecimal uses 16 as a base. The standard number system that we use is base 10, so there are 6 values we cannot use a single digit to represent. To compensate, letters A through F are used. 0 to 9 are the same for both decimal and hexadecimal. The values 10 through 15 change to letters. For example, 10 is the letter A and 15 is the letter F.

In this example, 154 will be converted to hexadecimal. 154 is first converted to binary which is '10011010'. This is done using the process described earlier. The next step is to divide the 8 bits of '10011010' into two 4 bit parts: '1001' and '1010'. Just like when converting numbers to and from binary, the power of 2 table will be used. As there are only 4 bits, we only use 8, 4, 2, and 1. The last step is just converting the 4-bit values to a decimal. In our example, the left bits, when added together will give the result of 9. Looking up this value in the table gives the same value of 9. So the left part is 9 in hexadecimal. For the right 4 bits, the binary gives the decimal value of 10. If I look this up in the table, this gives me a hexadecimal value of A. Finally, we combine the left and right sides together to get the final result of 9A.

To convert the other way, the process is reversed.

10:55 – The Windows calculator can be used to convert between different unit types. When the Windows calculator first opens, it will default to the "Standard" mode which does not provide the ability to convert to hexadecimal or binary. To change the view, select the view menu and then select the view "Programmer". The Programmer view allows for a quick and simple way to convert between different types of units.

To convert the original example of 154, type in 154 into the Windows Calculator. Notice when the value 154 is entered, below that the value is shown converted into, hexadecimal, Octal or binary. Octal is not covered in this video, however it uses a base of 8.



The Windows Calculator is capable of showing binary numbers up to 64bits. Showing this many bits makes it hard to read. To reduce the number of bits shown, in the bottom left of the Windows Calculator is the word “Qword”. If you press Dword, the screen display will change to Dword (Double Word), then Word (Word), and finally Byte (Byte). Pressing these buttons will cycle through the options changing the number of bits that are displayed. The bits displayed will be 64 for Qword, 32 for Dword, 16 for Word and 8 for Byte. One caveat when changing the display is that Windows will display the value as a signed value. So given the example of 154, when the display is changed to Byte, the value will change to -102. Signed values essentially means that one of the bits determines if the value will be positive or negative. When working with networking, you want to use unsigned values. Unfortunately, this option cannot be changed in Windows Calculator. In this example, since 154 requires only a byte, the display of Word will be used. Using the display Word means that when all 8 bits are displayed in a byte the value will not be displayed as a negative value.

Once the decimal value has been entered, to change it to hexadecimal, it is a simple matter of selecting the option Hex on the left side. When one of these options is selected on the left-hand side, whatever value is currently in the Windows Calculator will be converted. In this case the value of 154 is converted to the hexadecimal value of 9A.

Even though the binary value is shown in the Windows Calculator, if the “Bin” option is selected, this will change the number displayed to binary. If you want to change to different unit types, it is just a matter of pressing the required button. For example, to change the value to hexadecimal, press the button “Hex”. To change it back to decimal, press the “Dec” option.

See <http://YouTube.com/ITFreeTraining> or <http://itfreetraining.com> for our always free training videos. This is only one video from the many free courses available on YouTube.

## References

“Installing and Configuring Windows Server 2012 R2 Exam Ref 70-410” page 198

## Credits

Trainer: Austin Mason <http://ITFreeTraining.com>

Script: Phil Guld <http://philguld.com>

Voice Talent: HP Lewis <http://hplewis.com>

Companion Document: Phil Guld <http://philguld.com>

Website and Social Media Manager: Phil Guld <http://philguld.com>

Video Production: Kevin Luttman <http://www.KevinLuttman.com>

Quality Assurance: Brett Batson <http://www.pbb-proofreading.uk>