



Community portal  
Recent changes  
Help

Tools  
[What links here](#)  
[Related changes](#)  
[Special pages](#)  
[Printable version](#)  
[Permanent link](#)  
[Page information](#)

Log in

Page

Discussion

Read

View source

View history

Search bitrary

# Software Development : Endianness

## Software Development

Arabic numbers consist of digits, regardless of the base of the number. For example, the sentence, "The price of a brand new cruise ship is 792555463 dollars.", contains the number 792555463, where the digit 7 is the **most significant digit** and the digit 3 is the **least significant digit**. The digits can be indexed, starting from index 0. As the number, in this case, the price of the cruise ship, increases, the *NumberOfDigits(number)* is an [increasing function](#). Therefore from simplicity's point of view it makes sense to use the index 0 for the least significant digit in stead of using it for the most significant digit.

the number:	7	9	2	5	5	5	4	6	3
the indices:	8	7	6	5	4	3	2	1	0
	most significant digit								least significant digit

Probably the reason, why the Western world writes the least significant digit to the right and the number of digits increases to the left is that the Western world uses [Arabic numerals](#) and in Arabic text is written from right to left, not like in English, where text is written from left to right. To distinguish the situation, where the index 0 is assigned to the least significant digit, from the situation, where the index 0 is assigned to the most significant digit, the Computer science [has 2 terms](#) ([archival copy](#), [archive\\_org copy](#)): **Little-Endian** and **Big-Endian**.

the number in <b>Little-Endian</b> format:	7	9	2	5	5	5	4	6	3
the indices:	8	7	6	5	4	3	2	1	0
the number in <b>Big-Endian</b> format:	3	6	4	5	5	5	2	9	7

Numbers can be in any base, including base 2, in which case the digits are bits, often marked as 1 and 0. In the case of bytes the synonym for the term "bit endianness" is [Bit Numbering](#). In addition to the bit endianness multi-byte numbers also have [Byte Endianness](#). That is to say, a portable C/C++ program that reads 4B int values from a bytestream has to take to account, preferably only at build-time, that in theory the CPU may have one out of 4 configurations:

CPU bit endianness	CPU byte endianness
Little-Endian	Little-Endian
Little-Endian	Big-Endian
Big-Endian	Little-Endian
Big-Endian	Big-Endian

## Observations and Additional Notes

[Citation](#) ([archival copy](#), [archive\\_org copy](#)): *Big endian byte ordering has been chosen as the "neutral" or standard for network data exchange and thus Big Endian byte ordering is also known as the "Network Byte Order". Thus Little Endian systems will convert their internal Little Endian representation of data to Big Endian byte ordering when writing to the network via a socket. This also requires Little Endian systems to swap the byte ordering when reading from a network connection. Languages such as Java manage this for you so that Java code can run on any platform and programmers do not have to manage byte ordering.*

*It is important to observe Network Byte Order not just to support heterogeneous hardware but also to support heterogeneous languages.*