

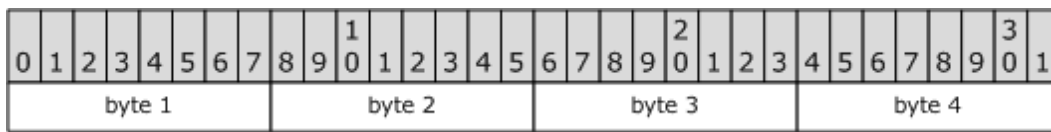
## 2.1 Common Base Types

This section contains commonly used primitive data types.

The use of the Interface Definition Language (IDL) implies RPC marshaling unless custom marshaling is specified.

Unless explicitly noted in this document, any integer, either signed or unsigned, is in memory order before RPC marshaling. It is implementation dependent [<1>](#) whether the memory order is **little-endian** or **big-endian**.

For packets, the bit numbering convention followed is the same as that used in RFCs, namely: the high bit of the first byte to hit the wire is in packet bit 0, and the low bit of the last byte to hit the wire is in packet bit 31 (so that the bits are shown from left-to-right in the order they naturally appear over the network).



**Figure 1: Packet byte/bit order**

So for example, on a little-endian system using a 32-bit field, the following table shows the correct correspondence between packet bit number and value on a little-endian system.

#	Network order value	Little-endian value	#	Network order value	Little-endian value
0	0x80000000	0x00000080	16	0x00008000	0x00800000
1	0x40000000	0x00000040	17	0x00004000	0x00400000
2	0x20000000	0x00000020	18	0x00002000	0x00200000
3	0x10000000	0x00000010	19	0x00001000	0x00100000
4	0x08000000	0x00000008	20	0x00000800	0x00080000
5	0x04000000	0x00000004	21	0x00000400	0x00040000
6	0x02000000	0x00000002	22	0x00000200	0x00020000
7	0x01000000	0x00000001	23	0x00000100	0x00010000
8	0x00800000	0x00008000	24	0x00000080	0x80000000
9	0x00400000	0x00004000	25	0x00000040	0x40000000
10	0x00200000	0x00002000	26	0x00000020	0x20000000
11	0x00100000	0x00001000	27	0x00000010	0x10000000
12	0x00080000	0x00000800	28	0x00000008	0x08000000
13	0x00040000	0x00000400	29	0x00000004	0x04000000
14	0x00020000	0x00000200	30	0x00000002	0x02000000
15	0x00010000	0x00000100	31	0x00000001	0x01000000