

# API Mode With Escaping

By Naveen PS



## API Mode

Determining which API mode to use can be confusing, on one hand using AP=1 option is straight forward to implement in software, whereas the AP=2 option seems to add only additional software complexity with little gain. The only difference between option 1 and 2 is the requirement to use escape characters. The AP=1 option eliminates escaping character sequences, and relies solely on the start delimiter and length bytes to differentiate API frames. If bytes in a packet are lost, the length count will be off, and the next API frame (packet) will also be lost. Digi has implemented the technique of escaping characters in an API frame in order to improve reliability, especially in noisy environments.

Using the AP=1 method can be easier to use, but does not provide the added reliability of using the AP=2, escape character sequence. For example, 0x7E is the flag character indicating the beginning of a new API packet. With the AP=2 option, since 0x7E can only appear at the start of an API packet, if 0x7E is received at any time it can be assumed that a new packet has started regardless of length. Certain control characters must be escaped if they are to be interpreted as data and not as the control characters themselves. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20. This allows the character to be passed as data. If the character is not followed by the escape sequence, the character is interpreted as a control character.

The following are all characters which need to be escaped:

**0x7E** Start frame delimiter.

**0x7D** Escape control character. Indicates that next byte is escaped.

**0x11 0x13** These bytes are software flow control characters.

Example:

Two XBee ZigBee nodes are configured for API=2 mode with escaped characters, all other parameters are default. From the transmitting radio the following raw Data Frame packet is sent:

0x7E      0x00      0x02      0x23      0x11      0xCB

The 0x11 (used by software flow control) byte needs to be escaped in this sequence. When the packet is received by another radios UART, the packet would look something like this:

0x7E      0x00      0x02      0x23      0x7D      0x31      0xCB

API packets are processed locally and only the data portion is sent over the air. If AP=2 option is enabled on the receiving radio, your host will have to convert the escaped data back into un-escaped information for processing.

Consider that radios set for transparent mode and radios set for AP=1 can communicate with radios set for AP=2. When the receiving modem is set to AP=2 it will always escape the control characters. Since only the payload portion of the API packet will be transmitted over the air, The receiving radios UART will alter the packet information base on its AP setting, allowing an AP=1 radio to successfully communicate with a radio set for AP=2.

# THE END

